

**Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Analýza vlivu detailu objektů v prostředí
Remcom Wireless InSite**

Level of Detail Analysis in Remcom Wireless InSite

Zadání diplomové práce

Student:

Bc. Lukáš Stryja

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2601T013 Telekomunikační technika

Téma:

**Analýza vlivu detailu objektů v prostředí Remcom Wireless InSite
Level of Detail Analysis in Remcom Wireless InSite**

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je navrhnout algoritmus pro automatické generování struktur objektů pro Wireless Insite s různou úrovní detailů a dále otestovat citlivost systému Wireless Insite na tyto různé úrovně. Součástí vypracování bude i rozbor a vyhodnocení výsledků.

1. Proveďte rešerši v oblasti popisu objektů pomocí různých datových formátů (např. KMZ, Collada) a v oblasti úrovně detailu (LOD - Level of Detail) pro tyto objekty.
2. Navrhněte algoritmus pro automatické generování struktur objektů s různým LOD (Level of Detail) pro Wireless Insite.
3. Otestujte sadou simulací vliv různých LOD na výkonnost a citlivost systému Wireless Insite.
4. Výsledky zhodnoťte.

Seznam doporučené odborné literatury:

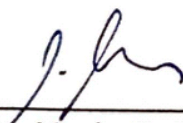
- [1] BILJECKI, F. *The Concept of Level of Detail in 3D City Models*. PhD Research Proposal, TU Delft, 2013, 70 pp. ISBN 978-90-77029-36-7
- [2] PHILLIPS, C. et al. *A survey of wireless path loss prediction and coverage mapping methods*. IEEE Communications Surveys & Tutorials, 2013, 15.1: 255-270.
- [3] Yun, Z., Iskander, M.F. *Ray tracing for radio propagation modeling: Principles and applications* (2015) IEEE Access, 3, art. no. 7152821, pp. 1089-1100.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Libor Michalek, Ph.D.**

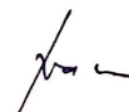
Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry






prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: *18. dubna 2019*

.....
podpis studenta

Poděkování

Rád bych tímto poděkoval svému vedoucímu, panu Ing. Liborovi Michalkovi, Ph.D. za čas a odbornou pomoc, kterou mi poskytl při řešení mé diplomové práce.

Poděkování patří také rodině za podporu při studiu, protože bez nich bych studium ani tuto práci určitě nedokončil.

Abstrakt

Tato práce se v teoretické části zaměřuje na popis objektů pomocí datových formátů, které podporuje komplexní softwarový nástroj pro predikci šíření elektromagnetického signálu Wireless InSite od společnosti Remcom. Dále se tato práce v teoretické rovině zabývá popisem v oblasti úrovně detailů (LOD - Level of Detail) a metodou sledování paprsků.

Praktická část se zabývá vytvořením algoritmu pro automatické generování struktur objektů školního kampusu VŠB-TUO s různým LOD (Level of Detail). Je zde také popsáno, jak došlo k vytvoření sady simulací, které testovaly vliv LOD na výkonnost a citlivost nástroje Wireless InSite. Nakonec bylo provedeno zhodnocení výsledků.

Klíčová slova

3D objekty; CityGML; Collada; DXF; impulzní odezva; KML; KMZ; LOD; metoda sledování paprsků; ShapeFile; STL; šíření signálu; úroveň detailů; vícecestné šíření; vlastnosti materiálů; Wireless InSite

Abstract

In the theoretical part, this thesis focuses on the description of objects using data formats, which supports the complex software tool Wireless InSite from Remcom, which is taught to predict electromagnetic signal propagation. Furthermore, this part in the theoretical level describes the LOD - Level of Detail and Ray Tracing method.

The practical part deals with the creation of algorithm for automatic generation of structures of school campus VŠB-TUO with different LOD (Level of Detail). It also describes how a set of simulations was created to test the impact of LOD on the performance and sensitivity of Wireless InSite. Finally, the results were evaluated.

Key words

3D objects; City GML; Collada; DXF; impulse response; KML; KMZ; Level of detail; LOD; material properties; multipath propagation; ray tracing method; ShapeFile; signal propagation; STL; Wireless InSite

Obsah

Seznam použitých symbolů.....	9
Seznam použitých zkratk.....	10
Seznam ilustrací a seznam tabulek.....	11
Úvod.....	14
1 Formáty pro ukládání objektů, se kterými umí Wireless InSite pracovat	16
1.1 DXF.....	16
1.2 Collada	17
1.3 KML.....	18
1.4 Shapefile.....	19
1.5 STL	21
1.5.1 STL formát v textové podobě.....	21
1.5.2 STL formát v binární podobě	22
2 LOD – Level of detail	24
2.1 Diskrétní LOD.....	25
2.2 Spojitý LOD.....	28
2.3 CityGML.....	28
3 Softwarový nástroj Wireless InSite.....	32
3.1 Objekty ve Wireless InSite.....	33
3.2 Ray Tracing metoda	35
3.2.1 Použití při simulacích šíření elektromagnetických vln.....	36
3.2.2 Typy paprsků (elektromagnetických vln).....	36
4 Vývoj algoritmu pro automatické generování struktur s různým LOD pro 3D prostředí ...	38
4.1 3D model kampusu VŠB – TUO	38
4.2 Popis algoritmu	40
4.3 Použité materiály.....	42
4.4 Vytvoření projektu ve Wireless InSite	45
4.5 Automatizace spouštění projektů	50
5 Vliv různých LOD na výkonnost a citlivost systému Wireless InSite	51
5.1 Otestování výpočetní náročnosti.....	51
5.2 Otestování citlivosti	53
5.2.1 Ukázkový projekt	53
5.2.2 Porovnání výsledků	56

Seznam použitých symbolů

Závěr	59
Použitá literatura	61

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
f	Hz	Frekvence
R_{xlev}	dBm	Přijímací úroveň
G	dBi	Zisk
σ	S/m	Konduktivita
ε_r	-	Relativní permitivita
l	m	Délka

Seznam použitých zkratk

Zkratka	Význam
2D	Dvojdimenzionální prostor
3D	Trojdimenzionální prostor
5G	Sít' páté generace
CPU	Central processing unit
CityGML	City Geography Markup Language
COLLADA	COLLABorative Design Activity
CAM	Computer Aided Manufacturing
Dae	Digital asset exchange
DCC	Digital content creation
DXF	Drawing Exchange Format
EM	Elektromagnetický
Shapefile	Esri Shapefile
XML	eXtensible Markup Language
GIS	Geografický informační systém
GDAL	Geospatial Data Abstraction Library
GPU	Graphic processing unit
HTML	Hypertext Markup Language
ITU – R	International Telecommunication Union - Radiocommunication Sector
KML	Keyhole Markup Language
KMZ	Keyhole Markup language Zipped
LOD	Level of detail
OGC	Open Geospatial Consortium
SCEA	Sony Computer Entertainment America
SIG 3D	Special Interest Group 3D
STL	standard triangle language
UML	Unified Modeling Language
WiFi	Wireless Fidelity
W3C	World Wide Web Consortium

Seznam ilustrací a seznam tabulek

Číslo ilustrace	Název ilustrace	Číslo stránky
1.1	DXF textový formát	17
1.2	Příklad KML 2.2 (element Placemark)	19
1.3	Jednoduchý příklad formátu Shapefile (body – vodní studny, křivky – řeky, mnohoúhelník – jezero)	20
1.4	Reprezentace CAD modelu prstence (naznačeného dvěma soustřednými červenými kružnicemi) a STL aproximace stejného tvaru (složeného z trojúhelníkových stěn)	21
1.5	Syntaxe STL v textové podobě. Anotace {}+ znamená, že tento blok se může opakovat	22
1.6	Syntaxe STL v binární podobě. Anotace {}+ znamená, že tento blok se může opakovat	23
2.1	Model busty s různou úrovní detailů (s různým počtem polygonů: 60 – 60 000)	24
2.2	Model panelového domu s různou úrovní detailů dle vzdálenosti objektu od kamery (číslo u budovy označuje počet použitých polygonů)	25
2.3	Model koule s různou úrovní detailů (s různým počtem vrcholů: 140 – 5 500)	27
2.4	UML diagram struktury modulů CityGML	29
2.5	Ukázka namodelované budovy pomocí sémantických celků založených na geometrických prvcích	29
2.6	Ukázka 4 úrovní detailů (LOD 1-4) v CityGML	30
2.7	Ukázka LOD 1 – 3D Model San Francisca vytvořený v platformě Smart City Explorer	31
2.8	Ukázka LOD 2 – 3D Model města Nimes ve Francii vytvořený v platformě Smart City Explorer	31
2.9	Ukázka LOD 3 – 3D Model města Bordeaux ve Francii vytvořený v platformě Smart City Explorer	31
3.1	Polygon	33
3.2	Podstruktura	34
3.3	Struktura	34

3.4	Skupina struktur	34
3.5	Kompletní 3D model	35
3.6	Struktura hierarchického geometrického uspořádání	35
3.7	Rozdíl mezi odraženým a difrakčním paprskem	37
4.1	Vizualizace kampusu VŠB-TUO v softwarovém nástroji Wireless InSite	40
4.2	Spuštění skriptu pro automatické generování struktur s různým LOD materiálu pro model kampusu VŠB-TUO	41
4.3	Pravděpod. výskytu materiálu 0 (betonu) je 100 %, 42 ostatních materiálů 0 %	
4.4	Pravděpod. výskytu materiálu 0 (betonu) je 0 %, ostatních 42 materiálů 100 %	
4.5	Vytvoření projektu ve Wireless InSite	45
4.6	Načtení 3D prostředí a terénu ve Wireless InSite	45
4.7	Charakter EM vlny	46
4.8	Boční pohled na 3D prostředí s rozvrženými přijímacími 47 body (Rx) a anténou (Tx)	
4.9	Nastavení studijní oblasti	48
4.10	Spuštění simulace	49
5.1	Graf ukazující čas potřebný ke zpracování simulací 51 v závislosti na různých LOD materiálu	
5.2	První sada simulací ve srovnání s druhou sadou simulací	52
5.3	Zpracování vícecestného šíření softwarovým nástrojem 54 Wireless InSite	
5.4	Časově proměnná impulzní charakteristika simulovaného 55 rádiového signálu	
5.5	Textový soubor obsahující 15 různých složek signálu, 55 které dorazily na přijímač v různých časových okamžicích (v sekundách) s různou přijímací výkonovou úrovní v jednotkách dBm	
5.6	Graficky znázorněné vícecestné šíření signálu	56
5.7	Graficky znázorněné jednotlivé složky signálu	58

Číslo tabulky	Název tabulky	Číslo stránky
4.1	Zvolené vlastnosti materiálů	42
4.2	Relativní permitivita a konduktivita změřená v konkrétních frekvenčních rozsazích dle ITU–R P.2040-1 (07/2015)	44
5.1	Porovnání výsledků 5 projektů s různým LOD	53
5.2	Význam zkratk při vícecestném šíření signálu	54
5.3	Porovnání jednotlivých složek signálu při různých LOD	57

Úvod

Valná většina lidí v dnešní době, ať už se podíváme kamkoli do světa, využívá bezdrátových technologií. Můžou to být mobilní technologie, abychom se dokázali spojit s kýmkoli a kdekoli. Můžou to být televizní technologie, které lidem zpřístupňují desítky až stovky televizních kanálů, na které se mohou během dne kdykoliv dívat. Můžou to být internetové technologie, jako například WiFi, které lidem dovolují získat informace na internetu o čemkoli, a nebojím se napsat, i o komkoli. Další bezdrátové technologie většina z nás využívá když platíme v obchodech bezkontaktními kartami nebo si jdeme odemknout auto. Zkrátka, ať chceme nebo nechceme, každého z nás se bezdrátové technologie nějakým způsobem týkají. A je jedno, jestli je používáme záměrně nebo nezáměrně, například když jdeme na vyšetření k doktorovi. Každý z nás s nimi přichází do styku. Málokdo si však uvědomuje, kolik úsilí a práce stojí oživit některou z výše jmenovaných technologií, či nějakou úplně novou. Nebo kolik času a prostředků stojí udržovat a zlepšovat již současné technologie. Lidé se nespokojují s tím co je, ale pořád chtějí více a více. Větší rychlost, větší kvalitu, větší zabezpečení – a to vše nejlépe za stejné peníze, nebo ještě menší.

K plánování, navrhování a optimalizaci bezdrátových sítí, odborníkům v této oblasti pomáhají sofistikované softwarové nástroje pro predikci šíření elektromagnetických signálů. Tyto nástroje jsou v mnoha případech schopny načítat modely měst a další 3D objekty a zasadit do těchto modelů anténní systémy a vyzkoušet různé bezdrátové technologie. Bývá velmi výhodné nejdříve zkusit dané technologie podrobit testům skrze tyto simulační nástroje. Může se tak ušetřit spoustu peněz, například tím, že se správně a na správná místa nainstalují potřebné systémy pro generování konkrétních elektromagnetických signálů. Nebo se zkrátka zjistí, že některé technologie v určitých místech použít nelze. Ano, tyto predikční simulační nástroje jsou pro technologie bezdrátových systémů velkým pomocníkem. Nejsou 100 % přesné, ale mohou hodně napovědět.

Jelikož i já si vybral pro své studium právě bezdrátové technologie, vznikla tato práce. V prvních třech kapitolách se zabývám spíše teoretickými poznatky k dané problematice. První kapitola se zabývá popisem objektů pomocí různých datových formátů, se kterými umí pracovat simulační nástroj Wireless InSite od firmy Remcom. Druhá kapitola se soustředí čistě jenom na různé úrovně detailů objektů (LOD – Level of detail). Zabývám se zde rozdílem mezi diskrétním a spojitým LOD a jejich možným využitím a nakonec zde popisuji otevřený datový model CityGML, který se využívá hlavně pro modelování infrastruktury ve více úrovních detailů. Kapitola 3 popisuje podrobněji softwarový simulační nástroj Wireless InSite a tato kapitola zahrnuje také podkapitolu o metodě sledování paprsků, známou spíše jako Ray Tracing metoda. Čtvrtá kapitola je celá praktická. Ze začátku se v ní lze dočíst, z čeho jsem vycházel, neboli, co všechno jsem měl k dispozici. Dále v ní popisuji 3D prostředí (3D model kampusu VŠB-TUO), které bylo využito při pozdějších simulacích a datový soubor nástroje Wireless InSite s příponou .OBJECT, který definuje právě 3D objekty a jejich vlastnosti. Následuje pak popis vytvořeného algoritmu pro automatické generování struktur s různým LOD pro 3D model školního kampusu. Další podkapitoly čtvrté kapitoly popisují použité materiály v budovách a jejich vlastnosti a sestavení jednoho projektu ve Wireless InSite a jeho následné naklonování ve 100 různých variantách. Poslední kapitola, pátá, rozebírá vliv různých LOD na výkonnost a citlivost nástroje Wireless InSite.

Tato práce si kladla za cíl prozkoumat, jak moc je potřeba řešit při simulacích bezdrátových technologií různé materiály, ze kterých jsou složeny budovy. Zda má smysl vytvářet propracované 3D modely infrastruktury a věrohodně kopírovat i všechny materiály, ze kterých jsou složeny, anebo stačí použít obyčejný model z jednoho konkrétního materiálu.

1 Formáty pro ukládání objektů, se kterými umí Wireless InSite pracovat

Tato kapitola se zabývá popisem datových formátů, se kterými dokáže aplikace Wireless InSite pracovat. Jedná se o tyto formáty:

- DXF
- COLLADA
- KML
- SHAPEFILE
- STL

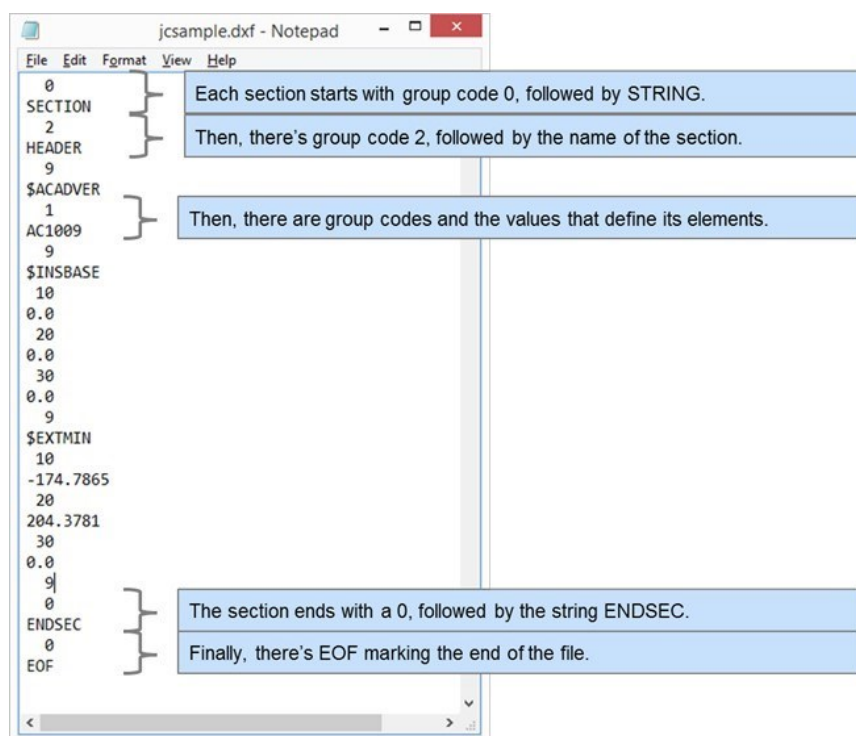
1.1 DXF

DXF (Drawing Interchange File) je jeden z nejpoužívanějších, ne-li nejpoužívanější vektorový grafický formát, ve kterém je možné ukládat jak informace o klasických 2D výkresech, tak i složité 3D modely, které jsou reprezentovány nejčastější trojúhelníkovými a čtyřúhelníkovými polygony, parametrickými plochami nebo také pomocí množinových operací CSG (Construction Solid Geometry). Tento formát byl navržen společností Autodesk, primárně pro světoznámou aplikaci této firmy – AutoCAD. Sekundární myšlenkou společnosti Autodesk bylo to, aby i další aplikace pro 2D či 3D modelování byly mezi sebou schopné komunikovat na souborové úrovni. V dnešní době (rok 2018) je vidět, že každá pokročilejší aplikace, která využívá 2D nebo 3D modelů, umí jak importovat, tak exportovat tento formát. [1]

Způsoby, jakými lze vektorová data uložit ve formátu DXF, jsou dva. Prvním a používanějším způsobem je uložit data jako text (nutno použít strukturovaný ASCII formát). Takový soubor je pak snadno čitelný nejen pro počítač, ale také pro člověka. Druhým, méně používaným způsobem je uložit data binárně. Výhodou je to, že výsledný soubor je přibližně o 20 % kratší (záleží na samotném souboru) a načítají se v aplikacích rychleji. Korektní přípona binárních souborů je .DXB. Někdy se však používá i méně korektní přípona .DXF, s kterou mohou mít některé aplikace či dokonce operační systémy problém, jelikož nerozeznají typ souboru podle této přípony. [1]

Hlavní plusy DXF formátu:

- Rozšířenost - Jedná se asi o největší výhodu tohoto formátu. Jak už bylo nastíněno v úvodu, i když byl formát DXF primárně určen pro aplikace pro 2D a 3D projektování a konstruování, tento formát souboru je v dnešní době podporován u většiny aplikací širokého zaměření, které pracují s 2D nebo 3D modely. Příkladem mohou být vektorové grafické aplikace, modelovací aplikace, fraktálové aplikace či aplikace jako je právě Wireless InSite, kde jsou 3D modely využívány při simulacích šíření elektromagnetického záření. [1]
- Struktura dat - Jednoduchá struktura má za následek u většiny DXF formátu snadné vytváření i zpětné načítání. Je to dané tím, že názvy atributů či příkazů jsou vždy uvedeny na lichých řádcích, kdežto samotné hodnoty jsou uvedeny na sudých řádcích (viz obrázek 1.1). Tím pádem nejsou potřeba žádné oddělovače ani žádné speciální knihovny, jak to někdy bývá u jiných datových formátů, které mají např. stromovou strukturu. [1]



Obrázek 1.1: DXF textový formát [2]

1.2 Collada

Collada nebo celým jménem COLLaborative Design Activity je otevřený standard pro sdílení digitálních aktiv, nejčastější 3D objektů a 3D animací. Na vzniku toho standardu se podíleli návrháři, vývojáři a další zájemci společnosti Sony Computer Entertainment America (SCEA), kteří tohoto formátu využívají zejména při tvorbě aplikací (nejčastěji her) pro konzoli Playstation a Playstation Portable. Na vytváření a rozšiřování tohoto standardu, jelikož se jednalo a stále se jedná o otevřený standard, se podíleli také návrháři a vývojáři z dalších významných společností jako: Alias Systems Corporation, Criterion Software, Autodesk, Inc. nebo Avid Technology. S vydáním verze 1.4.0 se COLLADA stal standardem Khronos Group Inc, kde se konsorciium pracující na tomto standardu rozhodlo nadále podporovat tento standard jako vrchol digitálního obsahu používaný ve 3D interaktivním průmyslu. [3][4][5]

Největší výhodou COLLADA standardu je, že je postaven na otevřeném XML schématu, které vývojářům usnadňuje vytvářet digitální obsah (DCC), kombinovat různé nástroje pro tvorbu a zpracování tohoto 3D obsahu (např. vizuálních scén včetně geometrie, shaderu, různých efektů, fyziky, animací, kinematiky anebo i několika rozdílných verzí interpretací stejných aktiv) a sdílet tento obsah mezi širokým spektrem aplikací. Toto sdílení se týká tzv. neutrální zóny v XML schématu, ve které vývojáři více firem mohou spolupracovat při navrhování společné specifikace pro nové nebo stávající aplikace či zařízení. Vývojáři konkrétních společností zde také mohou implementovat chování COLLADA formátu při importu a exportu v jejich dané aplikaci, kterou vyvíjí a tu poskytnout ostatním. Z výše uvedených faktů proto nemůže vzniknout propast v podobě nekompatibility mezi jednotlivými COLLADA formáty či aplikacemi. Proto je tento standard s oblibou používán u všech větších společností zabývajících se 3D obsahem. Jedná se například o společnosti vyvíjející tyto světoznámé aplikace: Maya, 3DS Max, Google Earth,

LightWave 3D, Maxon|Cinema 4D R11. Softimage|XSI, Side Effect's Houdini, MeshLab,, SketchUp, Blender, ArcGIS. [3][4][5]

Collada soubory bývají označeny příponou .dae, což je zkratka pro „digital asset exchange“, česky: Výměna digitálních aktiv.

1.3 KML

Keyhole Markup Language (KML) je geodatový formát založený na XML schématu vyvinutý společností Keyhole, Inc. pro použití v jejich aplikaci Earth Viewer. Tuto firmu však v roce 2004 koupila společnost Google, Inc. a Earth Viewer byl přejmenován na Google Earth. Důležitým milníkem v historii KML se stal den 16.4.2008, kdy se ve verzi 2.2 stal standardem OGC (Open Geospatial Consortium). [6]

KML je určený primárně k ukládání, spravování a zobrazování geografických dat v aplikacích, které umožňují:

- prohlížení Země nebo jejích částí jako ze satelitu,
- naklonění, přiblížení či oddálení konkrétní lokace v různých směrech,
- měření vzdáleností a rozlohu oblastí.

Pro lokalizaci se využívá souřadnicového systému WGS84 ve tvaru celých stupňů. Výškový systém je definován na základě elipsoidu WGS84 a geoidu WGS84 EGM96. [6]

2 typické aplikace, které využívají právě tohoto formátu jsou:

- Google Earth (virtuální glóbus)
- Google Maps (webový mapový server)

Existuje však celá řada dalších aplikací, které využívají KML pro prezentaci geografických informací. Příkladem může být webová vizualizace intenzity zemětřesení, dostupná na webových stránkách <https://earthquake.usgs.gov/> nebo třeba aplikace Wireless InSite, kterou se zabývá i tahle závěrečná práce. [6]

Základní elementy KML jsou tyto:

- Placemarks – Značka (párový tag *<Placemark>*), která označuje polohu na povrchu Země pomocí ikony žlutého špendlíku.. Každá značka obsahuje tři parametry:
 - Name – Jedná se o název (párový tag *<name>*), který se používá jako štítek pro značku.
 - Description – Jedná se o popis (párový tag *<description>*), který je zobrazen pod štítkem značky.
 - Point – Jedná se o popis konkrétní polohy značky (párový tag *<Point>*), pomocí zeměpisné délky, zeměpisné šířky a nadmořské výšky.

```
<?xml version='1.0' encoding='UTF-8'?>
<kml xmlns='http://www.opengis.net/kml/2.2'>
<Placemark>
  <name>FEI</name>
  <description>Fakulta elektrotechniky a informatiky.</description>
  <Point>
    <coordinates>18.1600711, 49.8309772,0</coordinates>
  </Point>
</Placemark>
</kml>
```

Obrázek 1.2: Příklad KML 2.2 (element Placemark)

- Descriptive HTML in Placemarks – U popisu značky lze využít HTML tagy pro upravení fontu písma (např. velikost, styl nebo barva), přidání odkazů a tabulek a nebo zarovnání textu.
- Ground Overlays – Tento objekt umožňuje překrýt namodelovaný terén obrázkem terénu. Obrázek musí být ve formátu .jpg a odkaz (či cesta k obrázku) musí být uvnitř párového tagu <Icon>.
- Path – Tento element (párový tag <LineString>), je tvořen vždy minimálně dvěma body, které obsahují souřadnice. Každé dva po sobě jdoucí body jsou propojeny čarou.
- Polygons – Tento element (párový tag <Element>) lze použít pro jednoduché vytvoření jakýchkoli objektů včetně celých budov.

Rozšiřující verzí souboru KML je soubor KMZ (Keyhole Markup language Zipped). Jedná se o komprimovanou verzi souboru KML. Tato komprimovaná verze obsahuje kořenový KML soubor doc.kml a soubory .jpg pro textury. Soubory KMZ lze otevřít například v aplikaci Google Earth nebo je rozbalit pomocí nástrojů pro de/kompresi jako je WinZip pro Windows, MacZip pro Macintosh a UnZip pro systémy UNIX. [6][8]

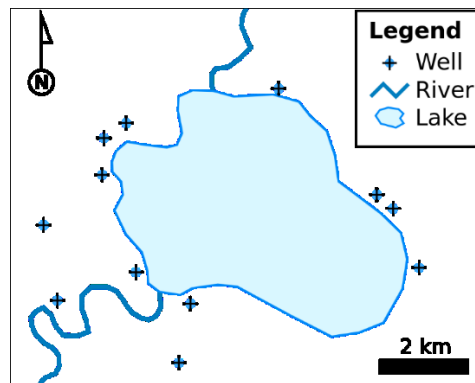
1.4 Shapefile

Shapefile nebo anglický též Esri Shapefile je oblíbeným geografickým vektorovým formátem pro aplikace geografického informačního systému (GIS). Byl vyvinut a stále je udržován společností Esri jako otevřená specifikace pro interoperabilitu dat mezi produkty Esri a dalšími aplikacemi GIS.

Shapefile formát umí prostorově popsat vektorové prvky:

- Body,
- křivky,
- mnohoúhelníky.

Tyto vektorové prvky reprezentují například vodní studny, řeky a jezera. Každé z položek jsou pak přidruženy určité parametry (vlastnosti) a po tomto přidružení vytvářejí reprezentaci geografických dat. U výše zmíněných vektorových prvků, by například parametry mohly být název a teplota. [9][10]



Obrázek 1.3: Jednoduchý příklad formátu Shapefile (body – vodní studny, křivky – řeky, mnohoúhelník – jezero) [9]

Shapefile se skládá z povinných a nepovinných doplňkových souborů. Povinně musí obsahovat následující 3 soubory:

- .shp – jedná se o hlavní soubor poskytující samotné vektorové prvky.
- .shx – jedná se o indexový soubor, který poskytuje poziční index vektorových prvků. Poziční index umožňuje rychlé vyhledávání vektorových prvků dopředu a dozadu.
- .dbf – jedná se o databázový soubor ve formátu dBase IV, kde jsou ke každému vektorovému prvku přiřazeny jeho parametry (vlastnosti).

Nepovinné soubory jsou tyto:

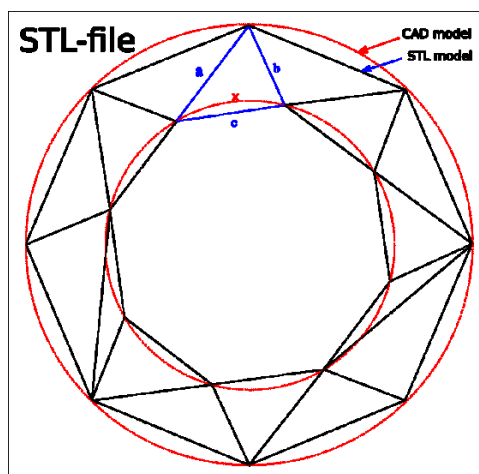
- .prj – jedná se o soubor poskytující informace o souřadnicovém systému a projekcích.
- .sbn a .sbx – jedná se o soubor obsahující seznam prostorových indexů parametrů prvků.
- .fbn a .fbx – jedná se o soubor obsahující seznam prostorových indexů parametrů prvků, které jsou pouze pro čtení.
- .ain a .aih – jedná se o soubor obsahující seznam indexu aktivních parametrů v tabulce.
- .ixs – jedná se o soubor obsahující seznam indexů geokódování pro čtení a zápis dat.
- .mxs – jedná se o soubor obsahující seznam indexů geokódování pro čtení a zápis dat (ODB formát).
- .atx – jedná se o soubor poskytující seznam indexů parametrů pro .dbf soubor
- .shp.xml – jedná se o soubor obsahující geoprostorové metadata ve formátu XML, podle zvoleného standartu.
- .cpq – jedná se o soubor popisující specifikaci kódování (pro soubor .dbf), pro správnou identifikaci znaků.
- .qix – jedná se o soubor obsahující alternativní seznam prostorových indexů parametrů prvků ve tvaru čtyřstromu (ang. Quadtree) pro aplikaci MapServer a aplikace postavené na knihovně GDAL (Geospatial Data Abstraction Library).

Názvy (předpony) u všech souborů by měly být složeny z 1-8 alfanumerických znaků (a-Z, 0-9), aby vyhovovaly i starším GIS aplikacím, které byly postaveny na konvenci pojmenování souborů 8.3 (8 znaků pro název, 3 znaky pro příponu). U novějších GIS aplikací je možno použít názvy delší. [9][10][11]

1.5 STL

Pro formát STL se používá mnoho jmen. Některé z nich jsou: „standard triangle language“, „stereolithography language“ nebo stereolithography tessellation language“. Chuck Hull ovšem uvádí, že zkratka STL vychází obecně ze slova „stereolitografie“. Chuck Hull je objevitel stereolitografie, tedy metody vytváření objektů pomocí postupného vytvrzování polymeru (nejčastěji tekuté světlocitlivé pryskřice) prostřednictvím záření různých vlnových délek, nejčastěji UV záření. Zároveň je spoluzakladatelem, viceprezidentem a vedoucím technologickým pracovníkem ve společnosti 3D Systems. Formát STL navrhl sdružení Albert Consulting Group pro firmu 3D Systems v roce 1987. [12] [13] [14] [15]

STL soubory popisují geometrii povrchu trojrozměrného objektu pomocí jednotkových normálových vektorů a vrcholů (uspořádaných podle pravidla pravé ruky), které tvoří trojúhelníky. Tento formát však nepopisuje barvy, textury nebo další obvykle parametry CAD modelu. [13]



Obrázek 1.4: *Reprezentace CAD modelu prstence (naznačeného dvěma soustřednými červenými kružnicemi) a STL aproximace stejného tvaru (složeného z trojúhelníkových stěn)* [13]

Formát STL se používá jak v textové tak v binární podobě. Použití té binární je však častější, protože jsou kompaktnější, a to tím více, čím je 3D model popsán STL souborem rozsáhlejší. [13]

1.5.1 STL formát v textové podobě

Každý STL soubor musí začínat řádkem:

```
Solid jméno
```

,kde *jméno* označuje název geometrického modelu a není povinné. Pokud však není zadáno, je nutno za *Solid* vložit mezeru. Soubor pak pokračuje popisem libovolného množství trojúhelníků. Každý z nich je nutno popsat následujícím způsobem:

```
facet normal  $n_i$   $n_j$   $n_k$ 
  outer loop
    vertex  $v1_x$   $v1_y$   $v1_z$ 
    vertex  $v2_x$   $v2_y$   $v2_z$ 
    vertex  $v3_x$   $v3_y$   $v3_z$ 
  endloop
endfacet
```

,kde každé n nebo v jsou čísla v reprezentaci s plovoucí desetinnou čárkou.

Příklad takového čísla: 5.471e-004. Soubor se ukončuje řádkem:

```
Endsolid jméno
```

solid *name*

$$\left\{ \begin{array}{l} \textbf{facet normal } n_i \ n_j \ n_k \\ \textbf{outer loop} \\ \textbf{vertex } v1_x \ v1_y \ v1_z \\ \textbf{vertex } v2_x \ v2_y \ v2_z \\ \textbf{vertex } v3_x \ v3_y \ v3_z \\ \textbf{endloop} \\ \textbf{endfacet} \end{array} \right\} +$$

endsolid *name*

Obrázek 1.5: Syntaxe STL v textové podobě. Anotace $\{ \}^+$ znamená, že tento blok se může opakovat. [16]

Formát STL umožňuje vytvářet objekty i pomocí dalších mnohoúhelníků. V praxi se však nejčastěji využívá skladba objektu pomocí trojúhelníků. [13]

1.5.2 STL formát v binární podobě

Využití binárního STL nachází uplatnění hlavně tehdy, pokud je STL soubor hodně rozsáhlý. STL soubor v binární podobě musí začínat 80 bytovou hlavičkou. Po ní musí následovat 32-bitové číslo bez znaménka pro uložení počtu trojúhelníkových stěn v souboru. Pak následuje zápis pro jednotlivé trojúhelníky. Každý trojúhelník je popsán dvanácti 32-bitovými čísly. První tři jsou určeny pro popis normály stěny a dalších 9 jsou určeny pro souřadnice X, Y, Z třech vrcholů trojúhelníku. Pak následuje 16 bitové celé číslo bez znaménka pro „attribute byte count“. Dokumentace však neuvádí, co přesně by mělo číslo „attribute byte count“ představovat. Je však uvedeno, že tato hodnota by měla být nastavena na hodnotu 0. Soubor je ukončen zápisem posledního trojúhelníku. [12] [13]

Bytes	Data type	Description
80	ASCII	Header. No data significance.
4	unsigned long integer	Number of facets in file
<div style="display: inline-block; vertical-align: middle; font-size: 4em; line-height: 1;">{</div>	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	4	float
	2	unsigned integer

i for normal
 j
 k
 x for vertex 1
 y
 z
 x for vertex 2
 y
 z
 x for vertex 3
 y
 z
 Attribute byte count

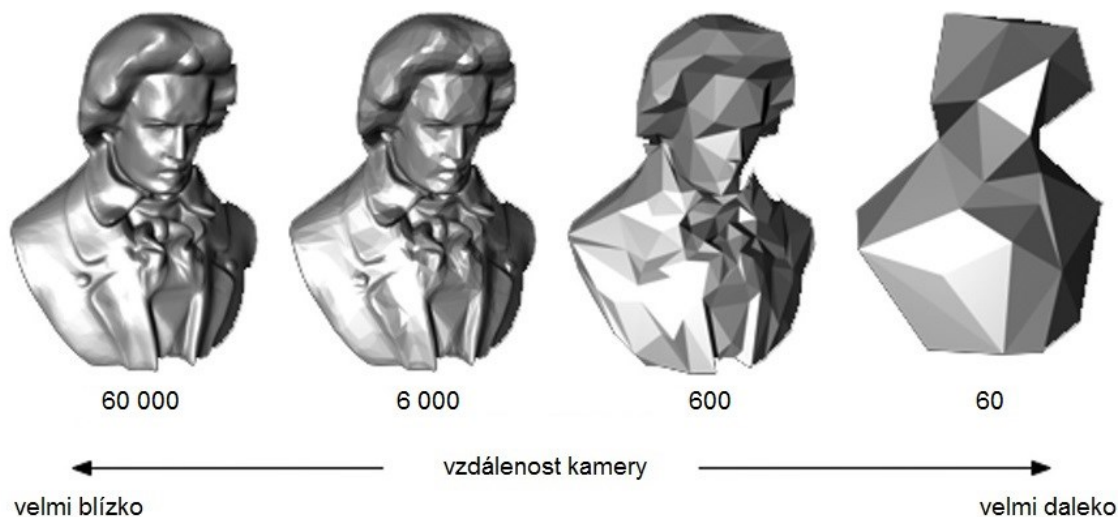
Obrázek 1.6: *Syntaxe STL v binární podobě. Anotace $\{ \}^+$ znamená, že tento blok se může opakovat. [16]*

STL formát našel uplatnění zejména při rychlém vytváření prototypů (rapid prototyping), 3D tisku a počítačové podpoře obrábění (CAM - Computer Aided Manufacturing). [13]

2 LOD – Level of detail

Řízení nebo také přepínání úrovně detailů (ang. Level of Detail, zkratka LOD) je velmi aktuálním (rok 2019) a zároveň velmi starým tématem v oblasti počítačové grafiky. Již v roce 1976 popsal James Clark přínosy zastupování objektů v rámci scény v několika rozlišeních. Vůbec poprvé se této techniky začalo využívat u leteckých simulátorů (80. léta 20. století), kde byly různé úrovně detailů objektů vytvářeny ručně. Už v té době bylo jasné, že LOD má budoucnost. Počátkem 90. let se pak díky rozmachu počítačů začala tato technika stále více využívat v počítačové grafice a vývoj nových algoritmů, dokumentů a softwarových nástrojů věnovaným vytváření a řízení úrovně detailu se od té doby po dnešní dobu nezastavil. [17]

LOD by šlo definovat jako techniku či disciplínu interaktivní počítačové grafiky, která se pokouší překlenout složitost a výkon tím, že reguluje množství detailů použitých k reprezentování virtuálního světa na základě vzdálenosti kamery (pohledu uživatele) od objektu [17]. Pro lepší pochopení této definice slouží obrázek 2.1. Na obrázku lze vidět jeden a ten samý model busty ve 4 variantách s různou úrovní detailů.



Obrázek 2.1: Model busty s různou úrovní detailů (s různým počtem polygonů: 60 – 60 000). [18]

Základním prvkem, který určuje úroveň detailu objektu je počet použitých trojúhelníků. V literatuře se však častěji používá namísto slova „trojúhelník“ termín „polygon“. Polygonem se sice obecně vzato označuje jakýkoli mnohoúhelník, ale v souvislosti s LOD se polygonem myslí právě trojúhelník a proto v této kapitole zůstávám u termínu „polygon“.

Na obrázku 2.1 bylo použito pro různou úroveň detailů objektu různé množství polygonů: přibližně 60 polygonů pro nejméně detailní variantu objektu až po přibližně 60 000 polygonů pro nejvíce detailní variantu stejného objektu. Je tedy pochopitelné, že rostoucí počet polygonů znamená detailnější zobrazení objektu. Pro počítačovou scénu platí, že čím je vzdálenost kamery (pozorovatele) od objektů menší, tím detailnější by daný objekt měl být. Naopak pokud bude kamera od objektu vzdálena velmi daleko, stačí použít méně detailnější objekt, protože lidské oko stejně nedokáže při dostatečně velké vzdálenosti (objekt bude velmi malý) postřehnout tak velké množství detailů. Touto technikou klesá hardwarová náročnost pro vykreslení počítačové scény.

Pokud by se měly všechny objekty vykreslovat v těch největších detailech, scéna by se nebyla schopna zobrazovat v rozumném čase. [17]

LOD lze rozdělit na dva základní typy:

- Diskrétní
- Dynamické

2.1 Diskrétní LOD

Diskrétní nebo též někdy nazývaný statický LOD je založen na tom, že již před spuštěním aplikace nebo části aplikace, která LOD využívá, se všechny dostupné úrovně detailů všech objektů nahrají do paměti a za běhu aplikace se pak na základě hardwarových prostředků a konkrétních kritérií rozhodne, která úroveň detailu se použije. [19]

Mezi konkrétní kritéria patří například tyto:

- Vzdálenost
- Velikost
- Priorita

1. kritérium – Vzdálenost

Jedná se o nejpoužívanější kritérium, podle kterého se přepínají různé úrovně detailu. Jak už bylo ukázáno na začátku této kapitoly, to jaká úroveň detailu se použije, záleží čistě na tom, jak daleko je kamera od objektu. Algoritmus, který přepíná jednotlivé úrovně, je pak například naprogramován tak, aby všechny objekty do 2 metrů byly zobrazeny v nejdetailnější podobě, zatímco objekty ve vzdálenosti od 2 do 5 metrů budou zobrazeny v o něco menší úrovni detailů, od 5 do 10 metrů ještě v menší úrovni detailů a od 10 metrů budou zobrazeny v nejméně detailní podobě. Největším úskalím při přepínání detailů pomocí tohoto kritéria je, kde zvolit v modelu ten správný bod, od kterého se bude počítat vzdálenost ke kameře. Na první pohled se může zdát rozumné, zvolit tento bod vždy doprostřed objektu. Je ale zřejmé, že v případě velkých objektů, jako jsou například stromy, budovy a podobné objekty, by toto umístění bodu nebylo moc efektivní. Je také nutno počítat s tím, že objekty se dle vzdálenosti zmenšují nebo zvětšují a proto je zapotřebí přistupovat ke každému objektu individuálně a vytvořit algoritmy, které vezmou při vkládání bodu v potaz výše uvedené informace. [20]



Obrázek 2.2: Model panelového domu s různou úrovní detailů dle vzdálenosti objektu od kamery (číslo u budovy označuje počet použitých polygonů). [21]

2. kritérium – Velikost

Přepínat mezi různými úrovněmi detailu lze také na základě velikosti objektů. Je jasné, že menší objekty nemusí být přepínány do nejvyšší úrovně detailů (není to však pravidlem), jelikož by tyto detaily lidské oko stejně nepostřehlo. U středních a velkých objektů je zase lepší zvýšit úroveň detailu, protože se uživatelé jeví tyto objekty více skutečné – připomínají realitu. Při počítání velikostí objektů se objekt (respektive všechny polygony) převedou do 2D plochy a teprve pak se spočítá skutečná velikost objektu. To si samozřejmě vybere svou daň na výpočetní náročnosti. [20]

3. kritérium – Priorita a lidské vnímání objektů

V některých aplikacích, zejména hrách se přepínání úrovní detailu řeší na základě tzv. priority a toho, jak člověk vnímá specifické objekty a jak často s nimi přichází do kontaktu. Pro příklad si představme hru, kde hráč ovládá svého bojovníka na koni a žene se vpřed proti nepřítelům. I když se hráč zrovna bude pohybovat vstříc nepřátelským jednotkám malebnou krajinou či po kamenité cestě, jeho zrak toto okolí nebude vnímat tak, jako nepřátelské jednotky, na které je teď zaměřen a přemýšlí jak je porazit. V tento moment je tedy možné zvýšit úroveň detailů nepřátelských jednotek, které jsou pro lidské oko prioritou a na které je hráč zaměřen. Naproti tomu prostředí, kterým hráč běží, ačkoliv je k němu blíže, není potřeba zobrazovat v těch nejlepších detailech, protože by tyto detaily hráči stejně unikly. [20]

Ke generování objektů v dané scéně se využívá dvou metod:

- Zdola nahoru
- Shora dolů






Metoda „zdola nahoru“

U této metody se bere ke zpracování nejdetaillnější úroveň objektů a jejím úkolem je zredukovat vrcholy či hrany jednotlivých polygonů, samotné polygony nebo i celé skupiny polygonů na nižší požadovanou úroveň detailů. Lze také přidávat další vrcholy, hrany či celé polygony, pokud to v konečném důsledku pomůže k redukci úrovně detailů původního objektu. Tato metoda je mnohem používanější, než opačná metoda „Shora dolů“. [19]

Metoda „shora dolů“

Při této metodě se využívá méně detailních objektů a využívá se techniky *subdivision*, která polygony původního objektu rozdělí na více nových polygonů a to přidáním vrcholů a hran do těch původních. Jedná se však o metodu, kdy se z jednoduššího modelu stává složitější a není tedy možné zaručit, že nová detailnější úroveň objektu bude ještě stále připomínat tu původní. [19]

Následující obrázek poslouží jako názorná ukázka obou dvou metod.

Image					
Vertices	~5500	~2880	~1580	~670	140
Notes	Maximum detail, for closeups.				Minimum detail, very far objects.

Obrázek 2.3: Model koule s různou úrovní detailů (s různým počtem vrcholů: 140 – 5 500). [19]

Pokud se na obrázek budeme dívat zleva doprava, lze říci, že se bude jednat o metodu „zdola nahoru“, kdy dochází ke zjednodušování původního modelu. Pokud se na tento obrázek budeme ale dívat zprava doleva, bude se jednat o metodu „shora dolů“, kde se skutečně povedlo z jednoduchého modelu udělat složitý. V praxi se však jedná o mnohem složitější a tvarem rozmanitější objekty, kde udělat z jednoduchého objektu složitý bývá samozřejmě o mnoho náročnější, než v tomto případě. Proto se tedy, jak už bylo psáno výše, využívá hlavně první metody „zdola nahoru“.

Výhody a nevýhody

Diskrétní LOD sebou přináší řadu výhod i nevýhod. Největší výhodou je fakt, že všechny objekty a jejich úrovně detailů se před startem aplikace před vypočítají, zoptimalizují a nahrají do paměti a za běhu aplikace již tento proces, který by mohl způsobit například zpomalení nebo sekání, není potřeba provádět. S tím souvisí i další výhoda a tou je samotné přepínání úrovní na základě popsaných kritérií na straně 25 a 26. Algoritmus, který se stará o přepínání objektů si hlídá jen určité hranice (jde v podstatě jen o přepínač, který přepíná různé úrovně detailu na základě několika podmínek) a podle toho vykreslí požadovanou úroveň objektu. Co se týče nevýhod, jsou známe 2 velké nevýhody tohoto diskrétního LOD. První nevýhodou je, že kvůli omezené velikosti paměti není možné vytvořit pro každý objekt například 20 různých úrovní detailů, ale například pouze 4 nebo 5, tudíž ne vždycky jde vykreslit úroveň detailu přesně takovou, jaká by byla zapotřebí, aby ji uživatel vnímal přirozeně. Druhou nevýhodou je tzv. přeblikávání, které je způsobeno tím, že se objekt nachází na pomezí 2 různých hranic. Představme si, že pro vzdálenost kamery od objektu od 0 do 2 metrů se všechny objekty budou vykreslovat v nejvyšší možné úrovni detailů. Pro vzdálenost kamery od objektu od 2 do 5 metrů však bude úroveň detailů o něco menší. Pokud se hráč bude pohybovat tak, že kamera od objektu bude vzdálena chvíli 1,9 metrů a chvíli 2,1 metrů a tento proces se bude stále opakovat dokola, bude mu daný objekt neustále blikat při hraniční hodnotě 2 metrů. Tento jev lze eliminovat některými metodami, avšak ne vždy se lze přeblikávání zbavit úplně. [20]

Metody pro eliminaci přeblikávání:

- Hysteréze – tato metoda přidává určitou toleranci k hranicím. Například pro výše uvedený příklad by se na základě neustálého přeblikávání rozhodlo o posunutí první hranice namísto 0 – 2 metrů na 0 – 2,5 metrů a druhá hranice by se posunula z 2 – 5 metrů na 2,5 – 5 metrů. [20]
- LOD blending – míchání sousedních LOD-ů je metoda, pomocí které se dva sousední modely kreslí pomocí míchání. [19]

- Inkrementální LOD – na objektech, u kterých dochází k problikávání se provedou jednoduché (elementární) přechody a přepnutí mezi danými objekty pak vypadá plynuleji. [19]

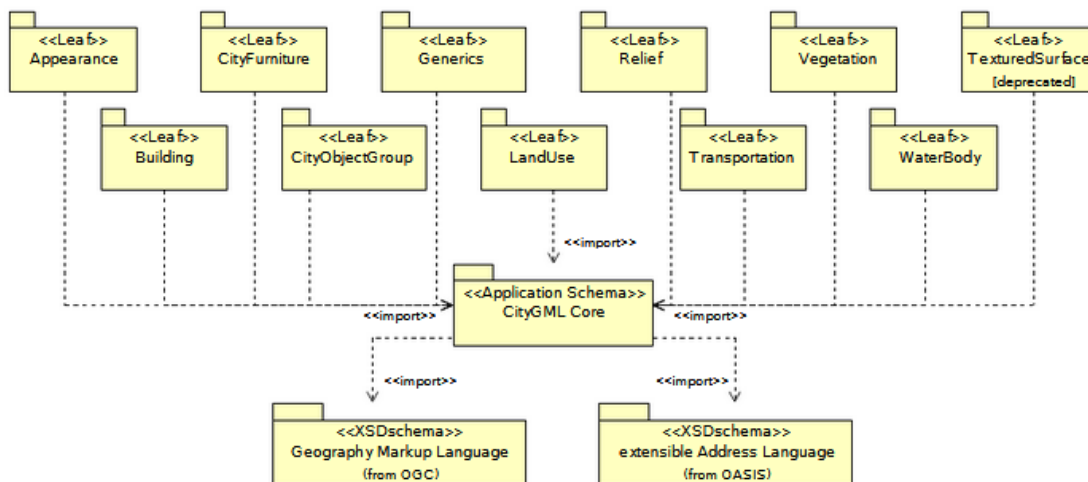
2.2 Spojitý LOD

Spojitý nebo též někdy nazývaný dynamický LOD je založen na opačném principu jako statický LOD. Modely se totiž nevytváří před spuštěním aplikace, ale až v jejím průběhu. I zde však existuje tzv. přípravná fáze, ve které dochází k uložení všech objektů do datové struktury. Z této struktury se pak vybírají modely na základě scény a jsou optimalizovány na úroveň detailů, která odpovídá věrohodné podobě objektu při použití co nejmenšího počtu polygonu. Výsledkem je tudíž lepší využití prostředků při zachování kvality objektů, která je pro lidské oko přijatelná. Další výhodou je eliminace problikávání. K tomu se využívá techniky zvané „progressive meshes“ nebo její vylepšené varianty „view-dependent PM“. Tyto techniky umožňují plynule volit úroveň detailů v závislosti na aktuálním pohledu. To znamená, že objekt se nejdříve vykreslí na základě toho, jak je uložen v datové struktuře a poté se např. na základě vzdálenosti plynule přidává nebo ubírá na detailech. Nevýhodou tohoto dynamického přístupu generování úrovně detailů je pochopitelně vyšší výpočetní náročnost, jelikož se objekty, jak už bylo napsáno, tvoří v reálném čase za běhu aplikace. [19][23]

2.3 CityGML

CityGML je aplikační schéma Geography Markup Language momentálně (leden/2019) ve verzi 3.1.1 (GML3) rozšiřitelného mezinárodního standardu pro výměnu prostorových dat vydaného konsorciem Open Geospatial Consortium (OGC) a ISO TC211. Dále se opírá o řadu norem z rodiny ISO 191xx, konsorcia W3C, webového 3D konsorcia a OASIS. Vývoj Specifikace CityGML byl zahájen v roce 2002 a první verze 1.0.0 byla vydána v říjnu roku 2008. Za tímto projektem stála skupina SIG 3D (Special Interest Group 3D). [24]

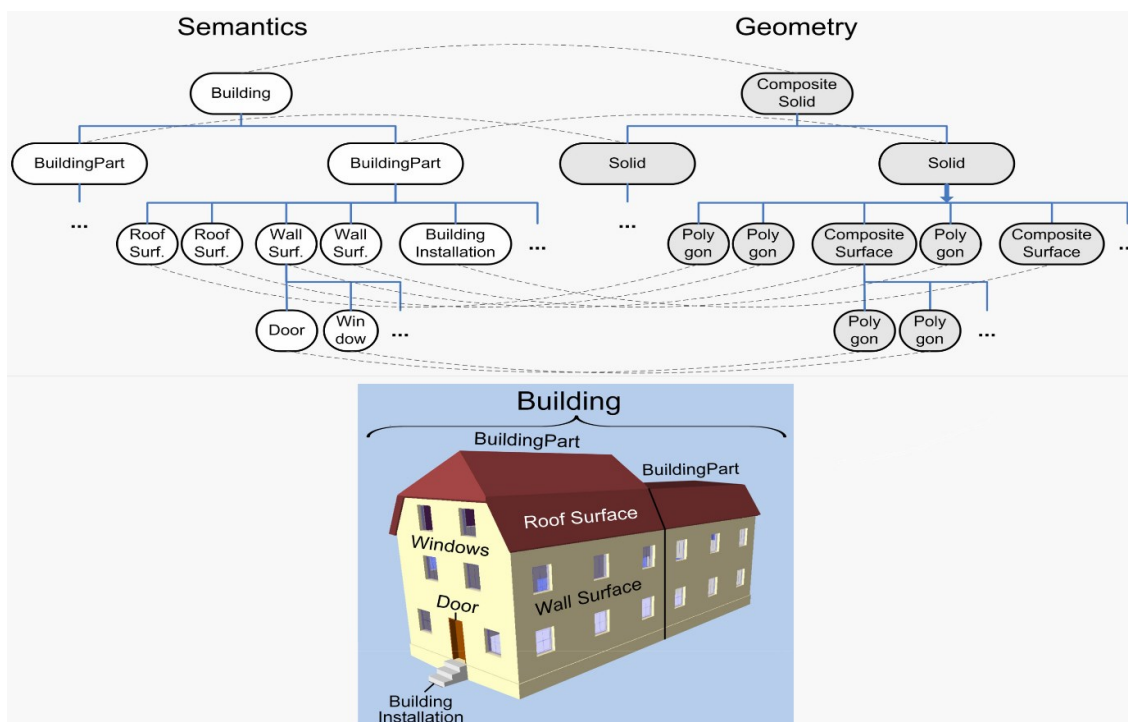
Jde o otevřený datový model založený na značkovacím jazyce XML, který slouží k reprezentaci a sdílení virtuálních 3D modelů měst nebo jejich částí. Tento datový model definuje třídy a vztahy pro nejrelevantnější topografické objekty ve městech s ohledem na jejich geometrické, topologické, sémantické a vzhledové vlastnosti. Jednoduše řečeno, tento model dokáže popsat různé třídy objektů, vztahy mezi nimi a s nimi související nevizuální i vizuální vlastnosti. Funguje zde také generalizace objektů, což je určitý proces výběru, zjednodušení a zevšeobecnění určitých prvků nebo částí objektů. [24]



Obrázek 2.4: UML diagram struktury modulů CityGML [24]

Sémantické a geometrické modelování

U CityGML se v dnešní době (rok 2019) využívá kombinace sémantického a geometrického modelování. V praxi to znamená, že 3D modely se skládají ze sémantických celků (např. roof surface, wall surface, door, window) a tyto sémantické celky jsou tvořeny geometrickými prvky – na nejnižší úrovni polygony. Následující obrázek ukazuje, jak jsou sémantické celky a geometrické prvky propojeny. [25]

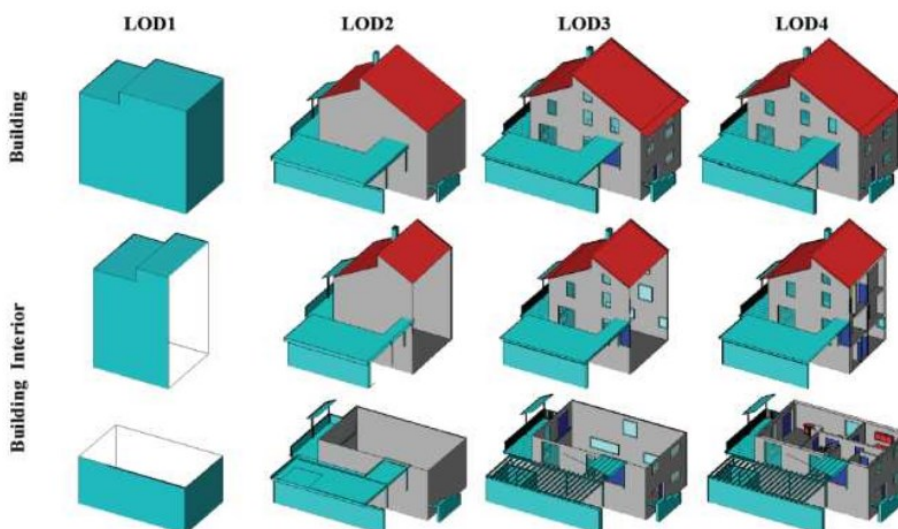


Obrázek 2.5: Ukázka namodelované budovy pomocí sémantických celků založených na geometrických prvcích. [25]

CityGML a LOD

3D objekty v CityGML – jedná se většinou o budovy – jsou děleny obvykle do 4 LOD úrovní (LOD 1-4). Ovšem někdy se lze v literatuře dočíst i o 5 LOD úrovních. V těchto případech se jedná o to, že je vytvořena ještě úroveň LOD 0, která reprezentuje 2,5 D model terénu a je logicky nejméně podrobná. Následující LOD 1-4 mají následující charakteristiku:

- LOD 1 – blokový model budovy bez střechy,
- LOD 2 – přidává k blokovému modelu střechu, terasu nebo například balkon. Zkrátka větší sémantické části, které patří k danému objektu,
- LOD 3 – doplňuje předchozí LOD o okna a dveře,
- LOD 4 – doplňuje předchozí LOD o interiér. Jak moc detailní je závisí na situaci a modelářích. [26]



Obrázek 2.6: Ukázka 4 úrovní detailů (LOD 1-4) v CityGML. [26]

Co se týče ostatních objektů, jako jsou například stromy, silnice, osvětlení a podobně, záleží vždy na situaci a velikosti objektu, kolik verzí LOD je zapotřebí nebo jak moc složitě bude objekt namodelován.

Využití

Virtuální modely měst se využívají nejen pro vizualizaci reálných měst nebo jejich částí, ale v konečném důsledku také pro sofistikované analýzy v různých aplikačních oblastech. Například společnost SIRADEL vyvíjí kolaborativní platformu Smart City Explorer s pokročilými funkcemi pro interaktivní 3D vizualizaci, simulaci a plánování městské infrastruktury. Na 3D modelech měst a jejich různých detailech může díky společnému uložení spolupracovat více firem, které mohou řešit různé části městské infrastruktury. Různými částmi městské infrastruktury se myslí například: Telekomunikační sítě, veřejné osvětlení, CCTV kamery, pozemní a podzemní doprava a mnoho dalších. Tyto důkladně propracované modely měst lze poté ukládat do formátů, s kterými umí pracovat další aplikace. V případě společnosti SIRADEL se jedná o využití těchto modelů v jejich aplikacích pro simulaci, vizualizaci a analýzu šíření elektromagnetických vln všech dnes používaných frekvencí pro různé bezdrátové technologie.



Obrázek 2.7: Ukázka LOD 1 – 3D Model San Franciska vytvořený v platformě Smart City Explorer. [28]



Obrázek 2.8: Ukázka LOD 2 – 3D Model města Nîmes ve Francii vytvořený v platformě Smart City Explorer. [28]



Obrázek 2.9: Ukázka LOD 3 – 3D Model města Bordeaux ve Francii vytvořený v platformě Smart City Explorer. [29]

Jednou z aplikací, kde se využívá 3D modelů objektů je i softwarový nástroj Wireless InSite, o kterém bude řeč v následující kapitole.

3 Softwarový nástroj Wireless InSite

Wireless InSite je komerční aplikace, která byla vytvořena a je stále vyvíjena (rok 2019) společností Remcom. Tato společnost od roku 1994 vyvíjí aplikace, které se zabývají šířením elektromagnetického signálu v nejrůznějších oblastech - od medicíny, přes automobilovou sensoriku až po nejnovější bezdrátové technologie jako je 5G. Jelikož se jedná o velice sofistikované aplikace, většina těchto aplikací vyžaduje náročnější hardwarové požadavky na procesor, grafickou kartu a operační paměť. Stojí také za zmínku, že většinu z těchto aplikací si může koncový zákazník po domluvě s personálním oddělením Remcomu na 30 dní vyzkoušet. Jde přitom o plné verze, takže koncový zákazník může před koupi s jistotou zjistit, zda mu daná aplikace bude opravdu vyhovovat nebo ne. Nevýhodou je, že společnost Remcom nevytvořilo pro své zákazníky internetové fórum, kde by mohli sdílet své nápady a postřehy a v případě nouze si poradit. U Remcomu se toto řeší pouze osobní konzultací. Ale zpět k aplikaci Wireless InSite. [30]

Wireless InSite je aplikace, která umí pracovat s různými 3D objekty (od malých objektů jako je nábytek nebo automobily přes samostatné vícepodlažní budovy po celá města plných budov). Tyto 3D objekty lze buďto přímo vytvářet v jednoduchém editoru této aplikace, což ale bývá velmi náročné a ne vždy použitelné, nebo je lze do aplikace importovat, což bývá většinou efektivnější a rychlejší varianta. Importovat 3D objekty lze ve formátech uvedených v kapitole 1. Hlavním účelem této aplikace ovšem není vytvářet, importovat a pracovat s 3D objekty. Hlavním účelem této aplikace je tyto 3D objekty umět použít při simulování šíření elektromagnetického vlnění na různých vlnových délkách (30 MHz až 300 GHz), a to buď ve vnějším nebo vnitřním prostředí. K simulování elektromagnetického šíření se ve Wireless InSite využívá několika známých modelů. Některé z nich jsou založeny na velmi přesné Ray Tracing metodě, o které bude řeč v podkapitole 3.2. Ty zbylé jsou založeny na méně sofistikovaných metodách, ale jejich výhodou je menší výpočetní čas. Zde je seznam všech dostupných modelů ve Wireless InSite:

- Full 3-D (podpora Ray Tracing metody, ale bez využití grafické karty)
- X3D (podpora Ray Tracing metody s využitím grafické karty (technologie CUDA))
- Urban Canyon
- Vertical Plane
- Real time – COST-Hata
- Real time – Freespace
- Real time – Hata
- Real time – OPAR
- Real time – Triple Path Geodesic
- Real time – VPUP
- Real time – Walfisch-Ikegami

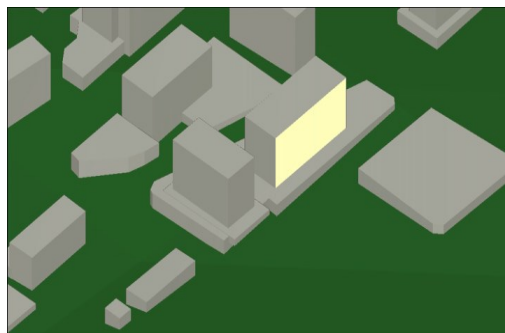
3.1 Objekty ve Wireless InSite

Modely objektů Wireless InSite rozděluje na 5 typů:

- Město – jedná se o nejsložitější typ. Jak již sám název napovídá, jedná se o souhrn 3D modelů budov, které tvoří město nebo část města.
- Terén – jedná se o profil terénu. Zde platí, že tento typ objektu musí být umístěn vždy pod všemi ostatními typy objektů
- Vegetace – jedná se o 3D oblasti, jako jsou lesy, husté parky nebo hory, které mohou být při simulacích velmi
- Vícepodlažní budova – jedná se o většinou velmi propracovaný 3D model vícepodlažní budovy, ve které je zahrnuto spoustu detailu (kanceláře s nábytkem, dveře a okna a další menší objekty).
- Samostatný objekt – jedná se o 3D modely nejrůznějších věcí. V předcházejícím bodě byl uveden například nábytek, ale může se jednat také o auta, telefonní budky, samostatné stromy a podobné objekty.

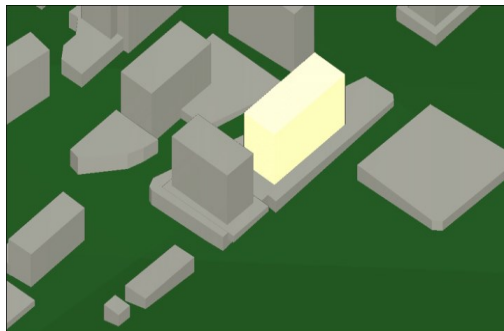
Každý z objektů má své hierarchické geometrické uspořádání. Toto uspořádání slouží k rozdělení objektů na menší části až po elementární prvky. Takové uspořádání je velmi výhodné, protože díky němu se objekt může skládat z různých materiálů o různých tloušťkách a může mít různé další vlastnosti. Jinak řečeno, nebýt hierarchického geometrického uspořádání, každý objekt by musel být celý vytvořen z jednoho materiálu se stejnými vlastnostmi. To by ubralo na věrohodnosti vypočtených výsledků při simulacích. Hierarchické geometrické uspořádání se skládá celkem z 5 úrovní.

- Polygon
 - Podstruktura
 - Struktura
 - Skupina struktur
 - Kompletní 3D model
- **Polygon** – Základ hierarchického geometrického uspořádání. Jde o ten nejjednodušší geometrický prvek. Polygon neboli také mnohoúhelník je tvořen třemi a více body propojenými do jednoho 2D útvaru. [31]



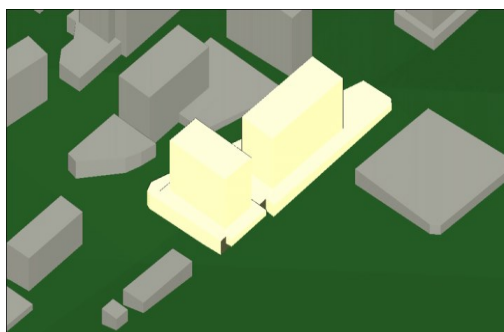
Obrázek 3.1: Polygon [31]

- **Podstruktura** — Uskupení většího množství polygonů. Aby se jednalo o podstrukturu, je potřeba zajistit, aby se hrany všechny polygonů dotýkaly a tvořily tak jeden logický 3D objekt. [31]



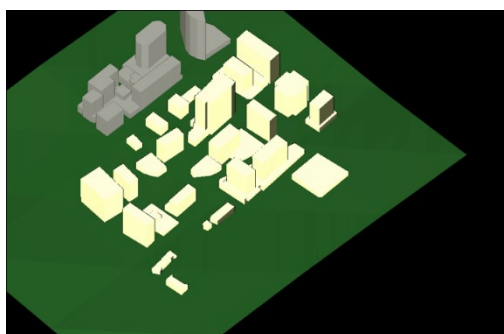
Obrázek 3.2: *Podstruktura [31]*

- **Struktura** — Kolekce podstruktur. Aby se jednalo o strukturu, musí se substrukтуры, které ji tvoří překrývat. Může nastat případ, kdy se podstruktura nebude překrývat s žádnou jinou podstrukturou. V tom případě se z podstruktury stane struktura. [31]



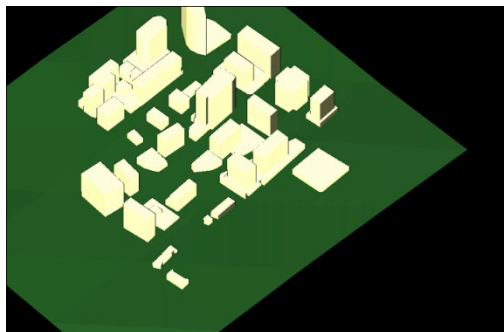
Obrázek 3.3: *Struktura [31]*

- **Skupina struktur** — Uskupení většího počtu struktur. Typickým příkladem jsou bloky nebo části města, specifické určitými společnými vlastnostmi. [31]



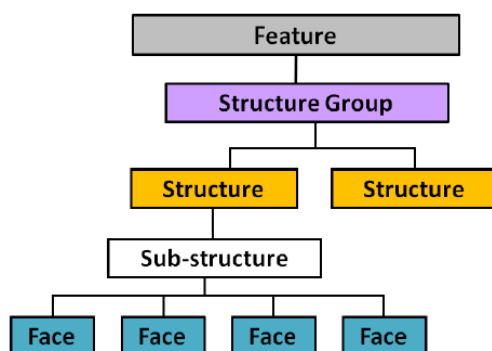
Obrázek 3.4: *Skupina struktur [31]*

- **Kompletní 3D model** – Kolekce více skupin struktur, jež tvoří kompletní 3D model připravený pro rozsáhlejší simulace. [31]



Obrázek 3.5: Kompletní 3D model [31]

Celá struktura hierarchického geometrického uspořádání je graficky znázorněna na obrázku 3.6.



Obrázek 3.6: Struktura hierarchického geometrického uspořádání [31]

3.2 Ray Tracing metoda

Ray Tracing metoda, česky nazývána jako „sledování paprsků“, je výpočetně vysoce náročná metoda, jejíž úkolem je na základě počítačových modelů napodobit, co nejvěrohodněji reálný svět – reálné předměty a scény (pochopitelně v 3D). [33]

Tato metoda umožňuje zobrazení odrazů, odlesků, stínů objektů a dalších jevů, což předchozí renderovací metody nedokázaly. Realistické počítačové obrazy vytvořené touto metodou nacházejí uplatnění v mnoha odvětvích. Například se ji využívá při tvorbě počítačových her a filmových efektů, v architektuře, ve vojenském a zdravotním průmyslu, při simulaci fyzikálních jevů nebo u virtuální reality. Jelikož je tato metoda, jak již bylo hned na začátku napsáno, výpočetně velmi náročnou metodou, v posledních letech se začíná znovu objevovat renderování pomocí GPU, jejichž architektury dovolují provádět daleko rychlejší výpočty, než CPU. [33]

Aby šla Ray Tracing metoda obecně použít, je zapotřebí splnit následující dvě podmínky:

1. Mít definovanou 3D scénu poskládanou z:

- 3D objektů, které mají svůj popis (tvar, pozici, materiálové vlastnosti)
- světelných zdrojů (pozici, barvu),
- dalších parametrů scény, jako je například barva pozadí nebo další vlastnosti daného prostředí

2. Mít definovanou aktuální pozici pozorovatele

Poté již stačí jen pozorovat a vyhodnocovat, co se děje s paprsky vyslanými ze světelných zdrojů. Jak se odrážejí, lámou a ohýbají na objektech. Na základě matematických modelů lze potom vykreslit realistickou scénu, podobnou realitě. [33]

3.2.1 Použití při simulacích šíření elektromagnetických vln

V případě, kdy však budeme řešit použití této metody při simulaci fyzikálních jevů, konkrétně šíření elektromagnetických vln, je potřeba jako v předchozím obecném případě zajistit dvě podmínky, kde první z nich je ovšem upravená:

1. Mít definovanou 3D scénu poskládanou z:

- 3D objektů (budovy, terén, vegetační pásma), které mají svůj popis (tvar, pozici, materiálové vlastnosti),
- zdrojů elektromagnetického signálu (nejčastěji antén),
- přijímačů elektromagnetického signálu (například mobily nebo rádia),

2. Mít definovanou aktuální pozici pozorovatele

Při použití Ray Tracing metody v tomto odvětví jsou zanedbatelné jevy jako stíny, odlesky a podobně, jelikož uživatele zajímá hlavně pokrytí dané oblasti konkrétním elektromagnetickým signálem nebo impulsní odezva daného systému. Využívá se však toho, jak již bylo výše naznačeno, co se s elektromagnetickou vlnou, respektive jedním impulsem stane, když narazí na překážku. To lze využít například k predikci signálu při vícecestném šíření EM vln. [33]

3.2.2 Typy paprsků (elektromagnetických vln)

Paprsek se ze zdroje (typicky antény) do cílového bodu (přijímače) může dostat několika způsoby. Podle toho paprsky dělíme na:

1. Přímý paprsek (Direct ray)

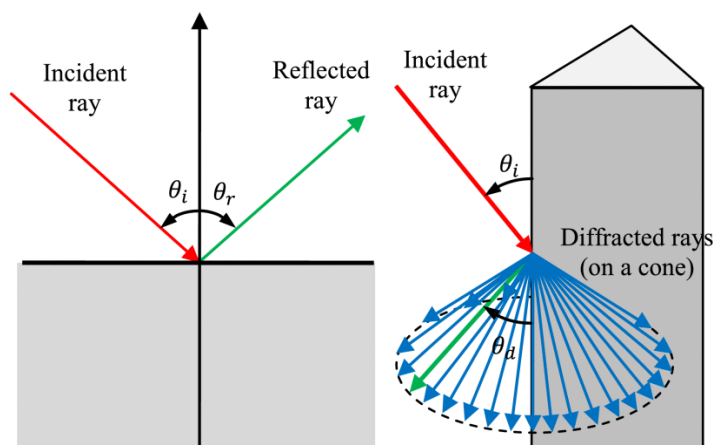
Jedná se o paprsek, který dorazí ze zdroje přímo do cíle, bez toho, že by kolidoval s nějakou překážkou.

2. Odražený paprsek (Reflected ray)

Jedná se o paprsek, který se při cestě ze zdroje do cíle jednou nebo vícekrát odrazil od překážky. Tento paprsek odpovídá odrazu elektromagnetických vln na rozhraní mezi různými materiálovými prostředími. Směr šíření odraženého paprsku je určen zákonem odrazu. Rozsah odraženého pole je určen Fresnelovými rovnicemi pro různé polarizace.

3. Difrakční paprsek (Diffracted ray)

Difrakční paprsky jsou oproti paprskům přímým a odraženým o mnoho složitější. Tyto paprsky se totiž při dopadu na jiné prostředí mohou „ohýbat“ do mnoha dalších směrů (nových paprsků) a s každým takto nově vzniklým paprskem je opět nutno počítat. Pro lepší pochopení slouží obrázek 3.7. Na levé straně tohoto obrázku lze vidět paprsek, který se odráží na rozhraní dvou prostředí podle zákona odrazu. Na pravé straně pak lze vidět paprsek a „ohybem“ nově vzniklé difrakční paprsky. Je jasné, že tento typ paprsků představuje velkou výpočetní náročnost.



Obrázek 3.7: Rozdíl mezi odraženým a difrakčním paprskem [34]

4. Rozptyl paprsků (Scattering)

Rozptyl paprsků, respektive rozptyl elektromagnetického záření je jev, kdy dopadající záření na rozptylující prostředí (vlnová délka je totožná či podobná s předmětem na který dopadá) mění své parametry, jako například směr, fázi nebo polarizaci. I zde je výpočetní náročnost mnohem vyšší, než například u odraženého paprsku. Ve Wireless InSite existují 3 modely, podle kterých lze rozptyl spočítat. Jsou to tyto:

- Lambertian Model – plně statistický nesměrový model
- Directive Model - směrový model, ale pouze s dopředným lalokem vzniklého rozptylu
- BackScatter Model – směrový model s dopředným i zpětným lalokem vzniklého rozptylu

Každý z těchto modelů používá jiný matematický aparát k tomu, aby mohly vzniknout difrakční paprsky na rozptylujícím prostředí.

4 Vývoj algoritmu pro automatické generování struktur s různým LOD pro 3D prostředí

Praktická část této práce se zaměřuje na vliv různých LOD stejného 3D modelu na výkonnost a citlivost profesionálního softwarového nástroje Wireless InSite, o kterém byla řeč v kapitole 3. Účelem bylo zjistit, jak velký smysl má řešit detaily při tvorbě 3D prostředí, které se následně používají při simulacích šíření konkrétních EM signálů. Zda má smysl zahrnovat do 3D modelů například skleněná okna, dřevěné dveře či další různorodé materiály konkrétních ploch na budovách nebo postačí tentýž 3D model avšak definovaný pouze jedním materiálem. Aby bylo možné výše uvedené otestovat, bylo kromě samotného softwarového nástroje Wireless InSite zajistit následující:

- 1) 3D prostředí neboli 3D model určité oblasti budov ve formátu, který zvládne Wireless InSite načíst.
- 2) Vytvořit algoritmus pro automatické generování struktur s různým LOD daného 3D prostředí.
- 3) Sestavit ve Wireless InSite 100 stejných projektů, které by se lišily pouze v různých LOD daného 3D prostředí.
- 4) Vytvořit Algoritmus pro automatické spouštění projektů ve Wireless InSite.

Následuje popis realizace těchto bodů.

4.1 3D model kampusu VŠB – TUO

Prvním úkolem bylo sehnat 3D model budov – neboli 3D prostředí, u kterého bych následně mohl měnit množství a různorodost materiálu a tyto různé varianty modelů mohl použít při simulacích. Zde bych rád poděkoval katedře geoinformatiky, která mi poskytla jednak kmz soubor reprezentující školní kampus (VŠB – TUO), tak soubor se stejnými daty, ale ve formátu *object*, s kterým umí pracovat Wireless InSite. Taktéž děkuji za poskytnutí terénu. Následuje ukázka souboru ve formátu *object* a jeho vysvětlení:

```

1      Format type:keyword version: 1.1.0      // Formát a verze
2      begin_<object> All                      // Počátek definice objektů
3      begin_<reference>                      // Počátek def. souř. systému
4      cartesian                              // Kartézský souř. systém
5      longitude 18.1563992610027             // Zeměpisná délka
6      latitude 49.8294600880967             // Zeměpisná šířka
7      visible yes                            // Viditelnost objektů povolena
8      sealevel                               // Souřadnice objektů počítány
                                           // od nadmořské výšky
9      end_<reference>                        // Konec def. souř. systému
10     begin_<Material> Concrete              // Počátek def. konkrétního materiálu
11     Material 0                             // Přiřazení čísla materiálu
12     LayeredDielectric                     // Materiál obsahuje dielektr. vrstvu
13     begin_<Color>                         // Počátek def. barvy materiálu
14     ambient 0.750000 0.820000 0.810000 1.000000
15     diffuse 0.750000 0.820000 0.810000 1.000000
16     specular 0.750000 0.820000 0.810000 1.000000
17     emission 0.000000 0.000000 0.000000 0.000000
18     shininess 5.000000
19     end_<Color>                            // Konec definice barvy materiálu
20     nLayers 1
21     begin_<DielectricLayer>               // Def. dielektrické vrstvy materiálu
22     conductivity 1.500e-002               // Konduktivita materiálu

```

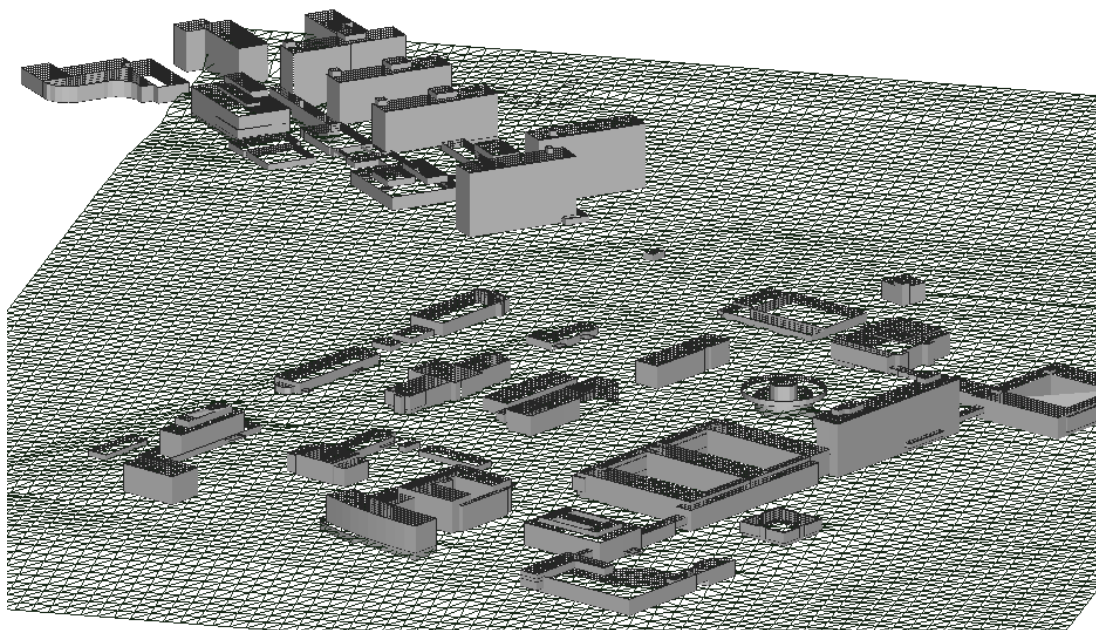
Vývoj algoritmu pro automatické generování struktur s různým LOD pro 3D prostředí

```

23      permittivity 7.000000          // Permitivita materiálu
24      roughness 0.000e+000          // Drsnost povrchu
25      thickness 3.000e-001          // Tloušťka materiálu
26      end_<DielectricLayer>          // Konec popisu dielektrické vrstvy
27      end_<Material>                 // Konec definice materiálu 0
28      begin_<Material> Metal          // Počátek definice materiálu 1
29      Material 1
30      PEC                            // Zkr. pro perfect electric conductor
31      thickness 0.000e+000          // Tloušťka vrstvy PEC
32      begin_<Color>
33      ambient 0.600000 0.600000 0.600000 1.000000
34      diffuse 0.600000 0.600000 0.600000 1.000000
35      specular 0.600000 0.600000 0.600000 1.000000
36      emission 0.000000 0.000000 0.000000 0.000000
37      shininess 75.000000
38      end_<Color>
39      roughness 0.000e+000
40      end_<Material>                 // Konec definice materiálu 1
41      begin_<Material> Asphalt_1GHz   // Počátek definice materiálu 2
      .
      .                               // Definice všech materiálů
      .
163     end_<Material>                 // Konec definice materiálu 7
164     begin_<structure_group>         // Počátek sk. struktur (viz str. 34)
165     begin_<structure>               // Počátek struktury (viz str. 34)
166     begin_<sub_structure>           // Počátek podstruktury (viz str. 34)
167     begin_<face>                   // Počátek polygonu (viz str. 34)
168     Material 0                     // Materiál polygonu
169     nVertices 6                     // Počet vrcholů polygonu
170     256.586116614344 515.516659138724 267.4 // Souřadnice 1. z 6. vrcholů
171     252.864116609213 508.094059125287 267.4
172     240.006116586446 514.372059139656 267.4
173     249.064116596594 532.484059165698 267.4
174     261.914116623171 526.144059157465 267.4
175     260.980216620548 524.281259158393 267.4 // Souřadnice 6. z 6. vrcholů
176     end_<face>                     // Konec polygonu
177     begin_<face>                   // Počátek dalšího polygonu
178     Material 0
179     nVertices 6
180     256.586116614344 515.516659138724 270.9
181     252.864116609213 508.094059125287 270.9
182     240.006116586446 514.372059139656 270.9
183     249.064116596594 532.484059165698 270.9
184     261.914116623171 526.144059157465 270.9
185     260.980216620548 524.281259158393 270.9
186     end_<face>                     // Konec polygonu
187     begin_<face>                   // Začátek dalšího polygonu
      .
      .
599306  end_<face>                     // Konec polygonu
599307  end_<sub_structure>             // Konec podstruktury
599308  end_<structure>                 // Konec struktury
599309  end_<structure_group>           // Konec sk. struktur
599310  begin_<ControlVectors>         // Definice řídících vektorů
599311  CVsVisible no
599312  Stippled no
599313  CVsThickness 3
599314  CVxLength 10.0000000000
599315  CVyLength 10.0000000000
599316  CVzLength 10.0000000000
599317  CVsXaxis 1.0000000000 0.0000000000 0.0000000000
599318  CVsZaxis 0.0000000000 0.0000000000 1.0000000000
599319  end_<ControlVectors>           // Konec Definice řídících vektorů
599320  end_<object>                   // Konec souboru

```

Vizualizaci tohoto souboru ve Wireless InSite ukazuje obrázek 4.1. Obecný popis tohoto souboru lze najít v referenčním manuálu Wireless InSite na straně 416 (příloha D.3) [31]



Obrázek 4.1: Vizualizace kampusu VŠB-TUO v softwarovém nástroji Wireless InSite

4.2 Popis algoritmu

Když již bylo zajištěno 3D prostředí, bylo zapotřebí navrhnout algoritmus, který by byl schopen s určitou mírou pravděpodobností měnit materiály jednotlivých polygonu, aby bylo možné otestovat, jaký vliv bude mít změna a množství různých materiálů na výsledky simulací a výpočetní náročnost.

Algoritmus pro generování struktur s různým LOD pro 3D model kampusu VŠB-TUO jsem vyvinul v jazyce Python, jelikož mi umožňoval napsat kód, který by dokázal efektivně prohledávat již stvořený soubor „object“ s 3D modelem kampusu VŠB-TUO a měnit pravděpodobnost výskytu jednotlivých materiálů.

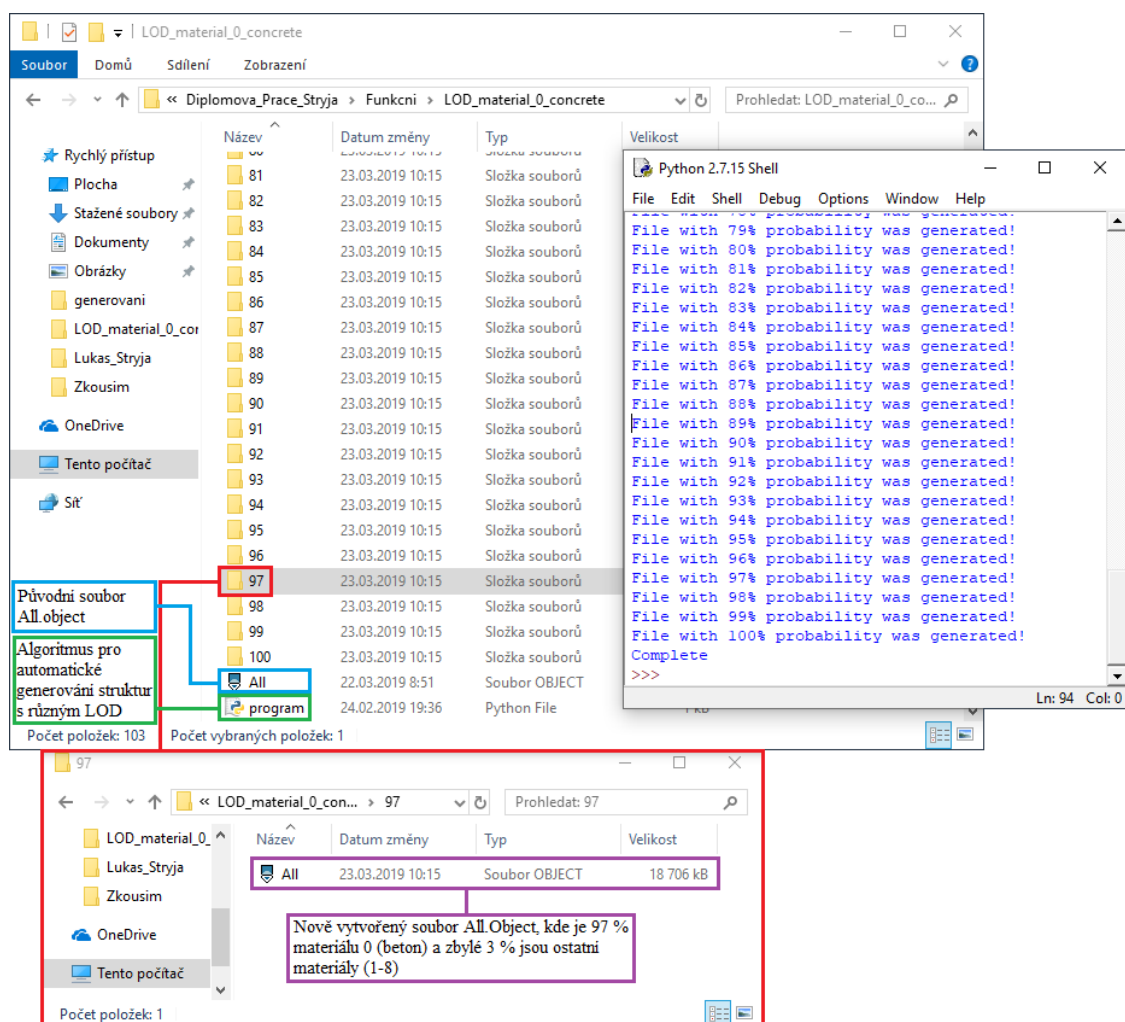
Algoritmus byl následující:

```
from random import randint, choice
from os import mkdir

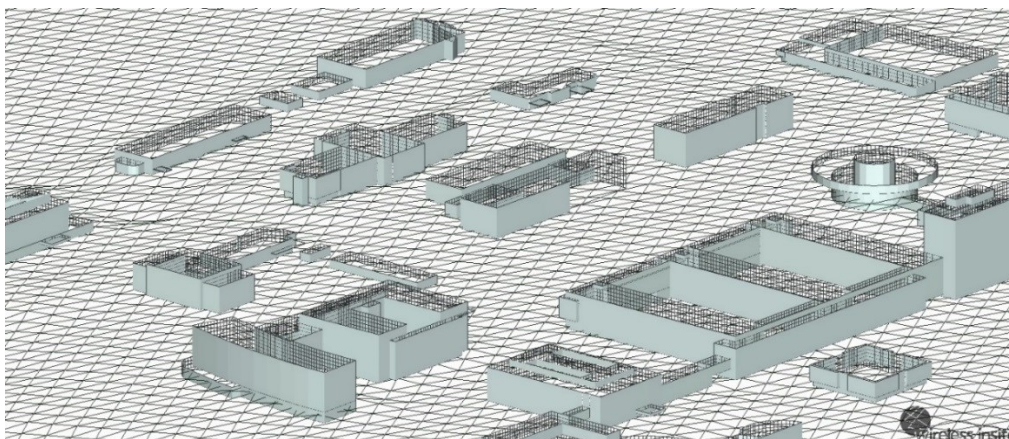
for randomValue in range(100 + 1):
    mkdir(str(randomValue))
    originalFile = open("All.object")
    newFile = open("{}\All.object".format(randomValue), "a")
    counterValue = 1
    for lineFromOriginal in originalFile.readlines():
        counterValue += 1
        if counterValue >= 169 and "Material" in lineFromOriginal:
            if randint(0, 100) <= randomValue:
                newFile.write("Material 0\n")
            else:
                newFile.write("Material{}\n".format(choice([1, 2, 3, 4, 5, 6, 7, 8])))
        else:
            newFile.write(lineFromOriginal)
    newFile.close()
    originalFile.close()
    print "File with {}% probability was generated!".format(randomValue)
print "Complete"
```


Algoritmus byl napsán tak, aby vytvořil 101 složek, od 0 po 100 a aby do každé z těchto složek vygeneroval jeden soubor s příponou .object. Soubory byly vytvořeny z již jednoho připraveného souboru (.object). To ukazuje obrázek 4.2.

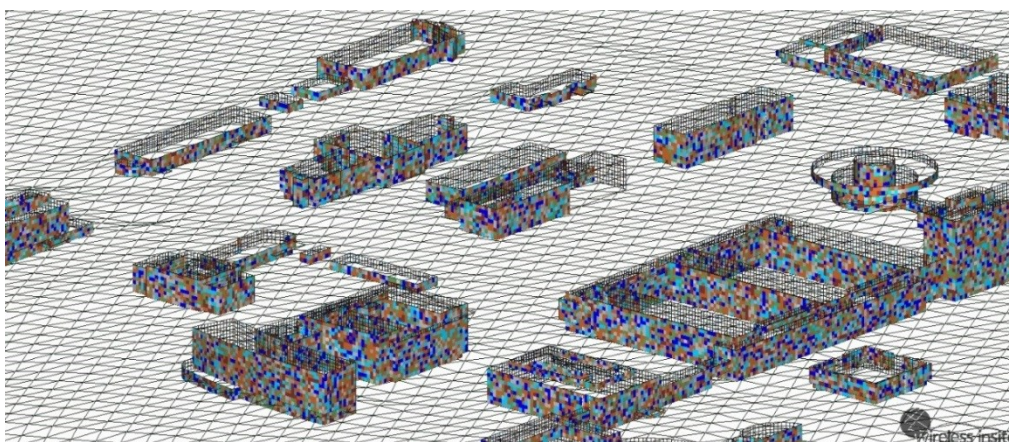
Program se na začátku každého cyklu (cyklus proběhne 101 krát) přesune na řádek 169 původního souboru All.object (na obrázku 4.2 označen modře) a odtud začne vyhledávat řetězec „material“. Jakmile ho najde, přiřadí za tento řetězec číslo od 0 po 8, které reprezentuje daný materiál a jeho vlastnosti. S každým cyklem, se zvyšuje pravděpodobnost (od nuly), že materiál s číslem 0 (beton) se tam objeví vícekrát. To znamená, že v prvním vygenerovaném souboru All.object je pravděpodobnost výskytu materiálu 0 (betonu) 0 %. Například v cyklu 50 se vygeneruje nový soubor All.object s pravděpodobností výskytu materiálu 0 (betonu) s 50 % pravděpodobností a v cyklu 100 se vygeneruje soubor .object s pravděpodobností výskytu materiálu 0 (betonu) se 100 % pravděpodobností. Na obrázcích 4.3 a 4.4 lze vidět nejjednodušší a nejsložitější variantu 3D modelu kampusu VŠB-TUO.



Obrázek 4.2: Spuštění skriptu pro automatické generování struktur s různým LOD materiálů pro model kampusu VŠB-TUO



Obrázek 4.3: Pravděpod. výskytu materiálu 0 (betonu) je 100 %, ostatních materiálů 0 %



Obrázek 4.4: Pravděpod.. výskytu materiálu 0 (betonu) je 0 %, ostatních materiálů 100 %

4.3 Použité materiály

Pro simulace bylo vybráno celkem 9 materiálů, které popisuje tabulka 4.1

Tabulka 4.1: Zvolené vlastnosti materiálů

Číslo	Materiál (anglicky)	Materiál (česky)	Tloušťka [m]	Relativní permitivita	Konduktivita [S/m]	Drsnost povrchu [μm]
0	Concrete	Beton	0,3	7	0,015	0
1	Metal	Kov	0,01	1	10000000	0
2	Asphalt	Asfalt	0,3	5,72	0,0005	0
3	Brick	Cihla	0,125	4,44	0,001	0
4	Fresh Water	Sladká voda	0,01	81	0,22	0
5	Glass	Sklo	0,003	2,4	0	0
6	Wood	Dřevo	0,03	5	0	0
7	FloorBoard	Podlah. deska	0,03	3,66	0,01437	0
8	Wet Earth	Mokrā zemina	0,1	25	0,02	0

Popis vlastností materiálů:

- Relativní permitivita ϵ_r
 - Bezrozměrná veličina, která udává poměr kapacity C kondenzátoru vyplněného konkrétním dielektrikem a kapacity C_0 kondenzátoru vakuového. Závisí na druhu dielektrika, na fyzikálních podmínkách (teplota, tlak) a pokud je kondenzátor zapojen do obvodu se střídavým proudem, tak závisí také na kmitočtu. Dielektrika mají relativní permitivitu vždy větší než 1, většinou 1-10. Feroelektrické látky ji mají v intervalu 100-1000. [35]
- Konduktivita σ
 - Též nazývaná měrná elektrická vodivost je fyzikální veličina, která popisuje schopnost konkrétní látky vést elektrický proud. O látkách, které mají vysokou hodnotu konduktivity lze říci, že jsou dobrými vodiči, kdežto o látkách, které mají konduktivitu nízkou lze říci, že jsou málo vodivé. Konduktivita závisí hodně na teplotě. To platí zvláště u polovodičů [36].
- Drsnost povrchu
 - Též nazývaná struktura povrchu nebo textura povrchu. Jde o termín, který vyjadřuje geometrické odchylky povrchu daného materiálu od jeho ideálního stavu. Drsnost povrchu vzniká při samotné výrobě materiálu (či výrobku) nedokonalostí strojů, které je vyrábějí. Geometrické odchylky mívají od jednotek po desítky mikrometrů [37].
 - Pokud se bavíme o elektromagnetických vlnách, právě na materiálech, které mají velkou drsnost povrchu, může vznikat tzv. rozptyl (ang. Scattering), které lze u některých systému dokonce využít v náš prospěch. V mnou provedených simulacích jsem ovšem u materiálů tento parametr zanedbal, respektive drsnost povrchu nastavil na hodnotu 0.

V případě, kdy jsou nám výše uvedené vlastnosti materiálů neznámé, ITU – R (Radiocommunication Sector of ITU) vydalo doporučení, označené jako ITU – R P.2040-1 (07/2015), které poskytuje některé důležité informace o materiálech, které se používají při stavbě budov nebo z kterých se skládá terén. Tabulka 2, která je zkopírovaná právě z tohoto doporučení poukazuje na změřenou relativní permitivitu a konduktivitu daných materiálu, při konkrétním rozsahu frekvencí.

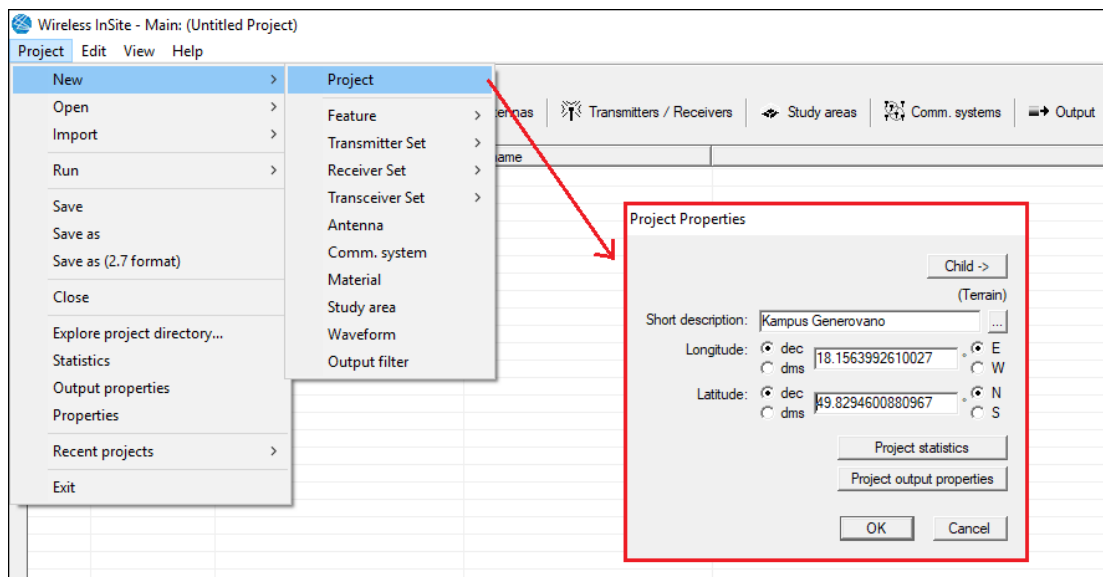
Tabulka 4.2: Relativní permitivita a konduktivita změřená v konkrétních frekvenčních rozsazích dle ITU – R P.2040-1 (07/2015) [38]

Material Properties					
Material class	Real part of relative permittivity		Conductivity S/m		Frequency range
	a	b	c	d	GHz
Vacuum (air)	1	0	0	0	0,001-100
Concrete	5,31	0	0,0326	0,8095	1-100
Brick	3,75	0	0,038	0	1-10
Plasterboard	2,94	0	0,0116	0,7076	1-100
Wood	1,99	0	0,0047	1,0718	0,001-100
Glass	6,27	0	0,0043	1,1925	0,1-100
Ceiling Board	1,5	0	0,0005	1,1634	1-100
chipboard	2,58	0	0,0217	0,78	1-100
Floorboard	3,66	0	0,0044	1,3515	50-100
Metal	1	0	10^{-7}	0	1-100
Very dry ground	3	0	0,00015	2,52	1-10
Medium dry ground	15	-0,1	0,035	1,63	1-10
Wet ground	30	-0,4	0,15	1,3	1-10

4.4 Vytvoření projektu ve Wireless InSite

Když už byly všechny soubory se 101 variantami úrovněmi detailu připraveny, bylo možné vytvořit ve Wireless InSite projekt a nastavit všechny potřebné parametry pro simulace. Tento projekt jsem pak 100x rozkopíroval a změnil pouze soubor s 3D prostředím, které se lišilo v úrovni detailu materiálů. Následuje popis vytvoření jednoho takového projektu ve Wireless InSite.

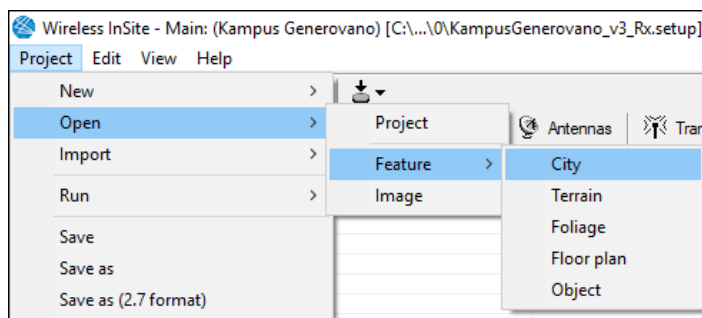
Vytvoření projektu



Obrázek 4.5: Vytvoření projektu ve Wireless InSite

Nahrání 3D terénu a prostředí

Načtení 3D terénu a prostředí do Wireless InSite je velmi jednoduché. Stačí nahoře kliknout na *Project -> Open -> Feature -> Terrain* a vybrat soubor s terénem a u 3D prostředí je to obdobné, je potřeba kliknout na *Project -> Open -> Feature -> City* a vybrat soubor, který obsahuje větší počet budov.

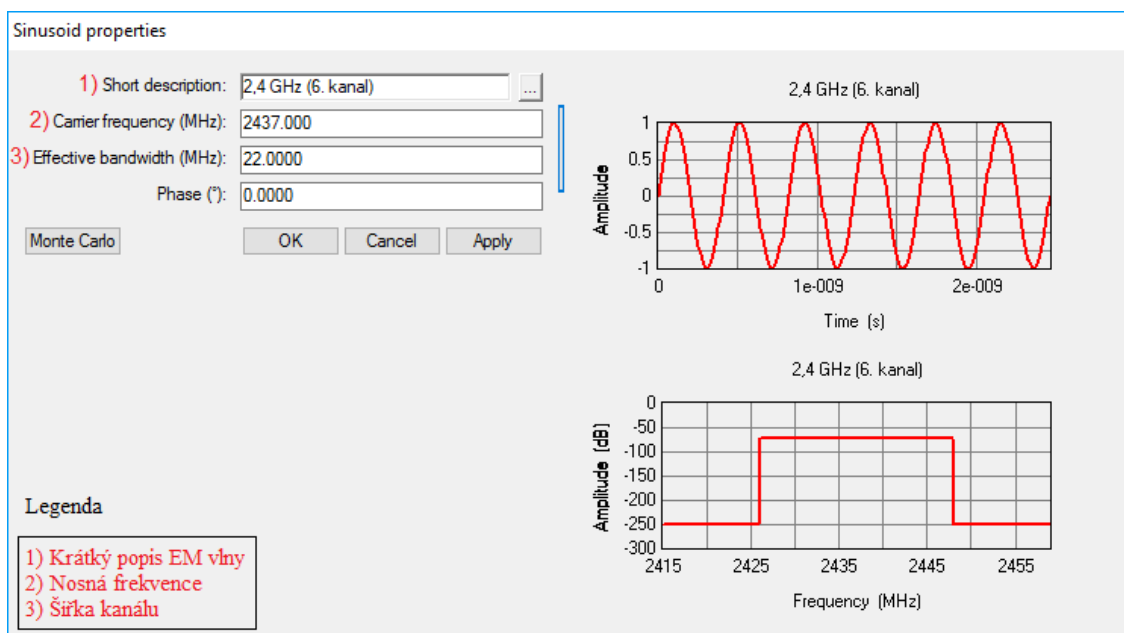


Obrázek 4.6: Načtení 3D prostředí a terénu ve Wireless InSite

To jak dané 3D prostředí a terén po načtení vypadal, lze vidět na obrázcích 4.3 a 4.4.

Použitá frekvence

Frekvence, kterou jsem pro simulace zvolil byla 2,4 GHz, jelikož se jedná o bezlicenční *pásmo ISM*. Parametry EM vlny zobrazuje obrázek 4.7.



Obrázek 4.7: Charakter EM vlny

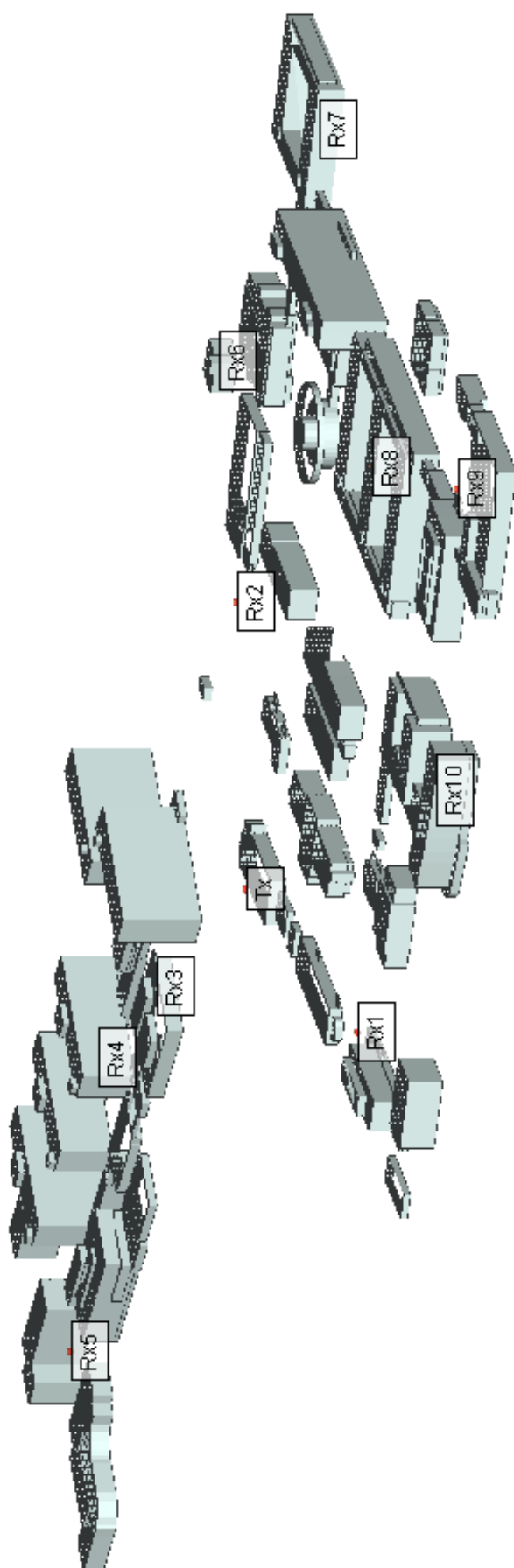
Přidání vysílací antény a přijímacích bodů do 3D prostředí

Nyní bylo nutno do nahraného 3D prostředí přidat vysílač a přijímací body (jako přijímací body si lze představit například mobilní telefony).

Protože se nejednalo o věrohodné napodobení žádné současné mobilní technologie (GSM, HSDPA, LTE atd.), parametry vysílače byly následující:

- Typ antény: Půlvlnný dipól
- Zisk: 6 dBi
- Poměr stojatých vln (PSV): 1
- Polarizace: Vertikální
- Vysílací výkon: 40 dBm

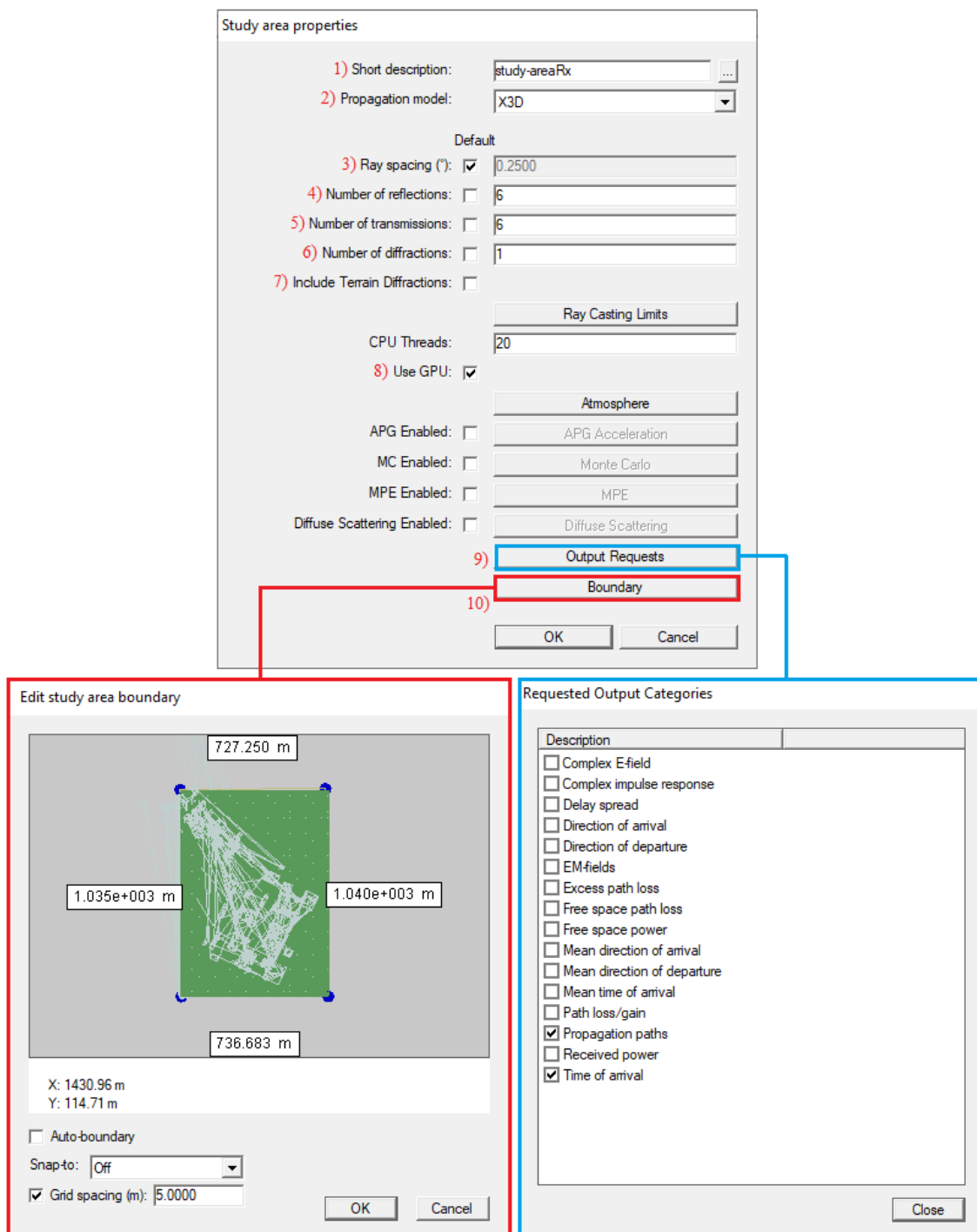
U přijímacích bodů bylo nastaveno pouze to, aby byly schopné zpracovat frekvenci 2,4 GHz a měli citlivost do -120 dBm. Přijímací body byly do 3D projektu rozmístěny na různá místa v různých výškách. Obrázek 4.8 zobrazuje přijímací body (Rx1 – Rx10) a anténu (Tx) položené v 3D prostředí.



Obrázek 4.8: Boční pohled na 3D prostředí s rozvrženými přijímacími body (Rx) a anténou (Tx)

Nastavení studijní oblasti

Nakonec bylo zapotřebí nastavit studijní neboli měřenou oblast, tedy oblast, kde se bude šíření EM vlny na frekvenci 2,4 GHz simulovat. Obrázek 4.9 zobrazuje nastavení studijní oblasti. Důležité položky jsou označeny červenými čísly a vysvětleny pod obrázkem.

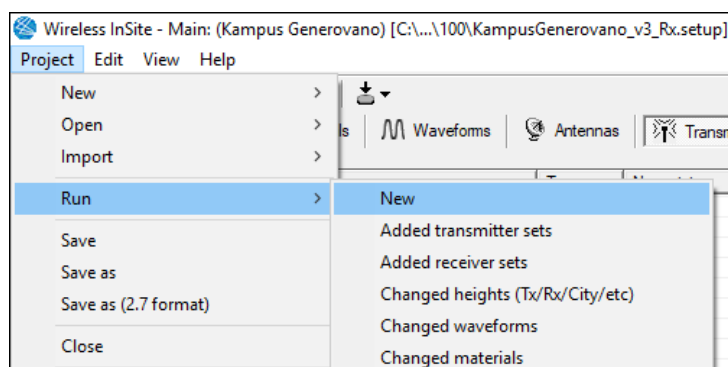


Obrázek 4.9: Nastavení studijní oblasti

Vysvětlení položek s červenými čísly z obrázku 4.9:

- 1) *Short description* – Krátký popis studijní oblasti.
- 2) *Propagation model* – Model šíření.
- 3) *Ray Spacing* (°) – Rozestup v úhlech při vyzařování paprsků.
- 4) *Number of reflections* – Počet odrazů (maximálně kolikrát se může EM vlna odrazit při jejím šíření).
- 5) *Number of transmission* – Počet přenosů (maximálně kolika překážkami může EM vlna v simulaci při jejím šíření projít).
- 6) *Number of diffraction* – Počet ohybů (maximálně kolikrát se může EM vlna v simulaci ohnout při jejím šíření).
- 7) *Include Terrain Diffraction* – Volba, zda zahrnout do simulaci i ohyby na terénu.
- 8) *Use GPU* – Volba, zda použít grafickou výpočetní jednotku.
- 9) *Output Requests* – Tato volba vyvolá nové dialogové okno. Vybírají se zde požadované výstupy simulací. V tomto případě byla vybrána cesta šíření paprsku od zdroje k cíli (Propagation paths) a impulzní odezva systému (Time of arrival)
- 10) *Boundary* – Tato volba vyvolá nové dialogové okno. Zde se určuje oblast, která se bude simulovat. V praxi to znamená, že se nemusí vždy simulovat celý projekt, ale jen jeho část.

Po kliknutí na tlačítko OK, se nastaví studijní oblast a je možné pustit simulaci. To by se provedlo tak, jak ukazuje obrázek 4.10



Obrázek 4.10: Spuštění simulace

Po dokončení nastavení projektu jsem tento projekt rozkopíroval přesně 101 krát do složek s názvem „0“ – „100“. V každém projektu jsem pak změnil soubor s různou úrovní detailu. Z důvodu, že projektů, které jsem potřeboval od simulovat bylo 101, napsal jsem dávkový soubor, který spouštění projektu z automatizoval.

4.5 Automatizace spouštění projektů

Jak již bylo napsáno v předchozí podkapitole, spouštět 101 projektů ručně a vytipovat, jak dlouho bude daná simulace trvat, aby bylo možné pustit další simulaci, by bylo časově velmi náročné. Napsal jsem tedy dávkový soubor pro Windows, který spouštění projektů jeden po druhém z automatizoval:

```
@echo off
Setlocal EnableDelayedExpansion

FOR /L %%X IN (0, 1, 100) DO (
    echo Projekt c. %%X
    echo Projekt c. %%X >> zaznam_z_vypoctu.txt
    echo Zacatek vypoctu: !time!
    echo Zacatek vypoctu: >> zaznam_z_vypoctu.txt
    echo !time! >> zaznam_z_vypoctu.txt
    wibatch.exe -f C:/Diplomova_Prace_Stryja/Funkcni/LOD_material_0_concrete/
    %%X/KampusGenerovano_v3_Rx.study-areaRx.xml -out C:/Diplomova_Prace_Stryja/
    Funkcni/LOD_material_0_concrete/%%X/study-areaRx
    echo Konec vypoctu: !time!
    echo Konec vypoctu: >> zaznam_z_vypoctu.txt
    echo !time! >> zaznam_z_vypoctu.txt
    echo.
    echo. >> zaznam_z_vypoctu.txt
    timeout 5
)

pause
```

Algoritmus funguje velmi jednoduše. Celý kód je ve smyčce FOR po dobu 101 cyklů. V každém cyklu se запиše do externího souboru (*zaznam_z_vypoctu.txt*) číslo projektu, čas začátku spuštění simulace a čas dokončení simulace. V každém cyklu se taktéž spouští příkaz *Wibatch.exe*. Tento příkaz spouští v každém cyklu jeden projekt (simulaci) a zapisuje její výsledky do podsložky *study-areaRx.xml*, která je obsažená vždy ve složce, odkud projekt pocházel.

5 Vliv různých LOD na výkonnost a citlivost systému Wireless InSite

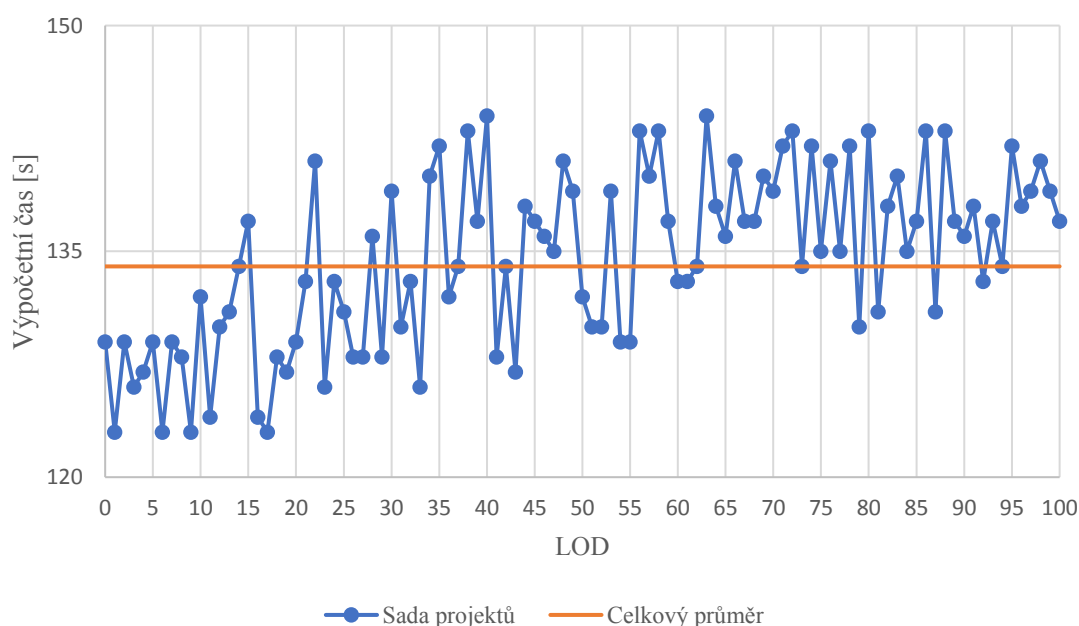
V této kapitole se zaměřuji na samotné výsledky simulací. Jak již bylo uvedeno v úvodu, cílem této práce bylo zjistit, jak velký vliv bude mít rozmanitost a počet různých materiálů (LOD) v 3D modelu testovaného prostředí na výkonnost a citlivost softwarového nástroje Wireless InSite od společnosti Remcom. Jinými slovy řečeno, zda má smysl vytvářet 3D propracované modely s věrohodným umístěním a typem materiálů, tak jak je to ve skutečnosti, nebo stačí jednoduchý 3D model z jednoho konkrétního materiálu.

V první podkapitole se zaměřuji na výkonnost systému, z pohledu potřebného času pro výpočet simulací v závislosti na různých LOD materiálu. V druhé podkapitole se věnuji citlivosti systému skrze porovnání impulzních charakteristik radiového kanálu, taktéž v závislosti na různých LOD materiálu.

5.1 Otestování výpočetní náročnosti

Jak již bylo zmíněno v úvodu této kapitoly, prvním aspektem, na který jsem se zaměřil, byl vliv LOD materiálu na výkonnost softwarového nástroje Wireless InSite. O výpočty simulací se starala velmi výkonná grafická karta NVIDIA Tesla m60-8A s podporou technologie CUDA.

Výsledky lze vidět na následujícím grafu (obrázek 5.1). Ten ukazuje, že z počátku byl čas pro zpracování simulací menší, jelikož první projekty obsahovaly nízkou úroveň detailů (LOD) materiálů. Postupně se však výpočetní čas zvyšoval, jelikož modely byly čím dál tím složitější. Zdá se však, že v druhé polovině grafu již byl výpočetní čas víceméně podobný a různé LOD materiálu už neměly na výpočetní čas takový vliv, jaký měly na začátku grafu.



Obrázek 5.1: Graf ukazující čas potřebný ke zpracování simulací v závislosti na různých LOD materiálu

Zvýšení počtu možných odrazů a přenosů

Z hlediska výpočetní náročnosti jsem se rozhodl otestovat ještě jednu věc. Zajímalo mě, o kolik vzroste výpočetní náročnost, když zvýším počet možných odrazů a přenosů elektromagnetického signálu.

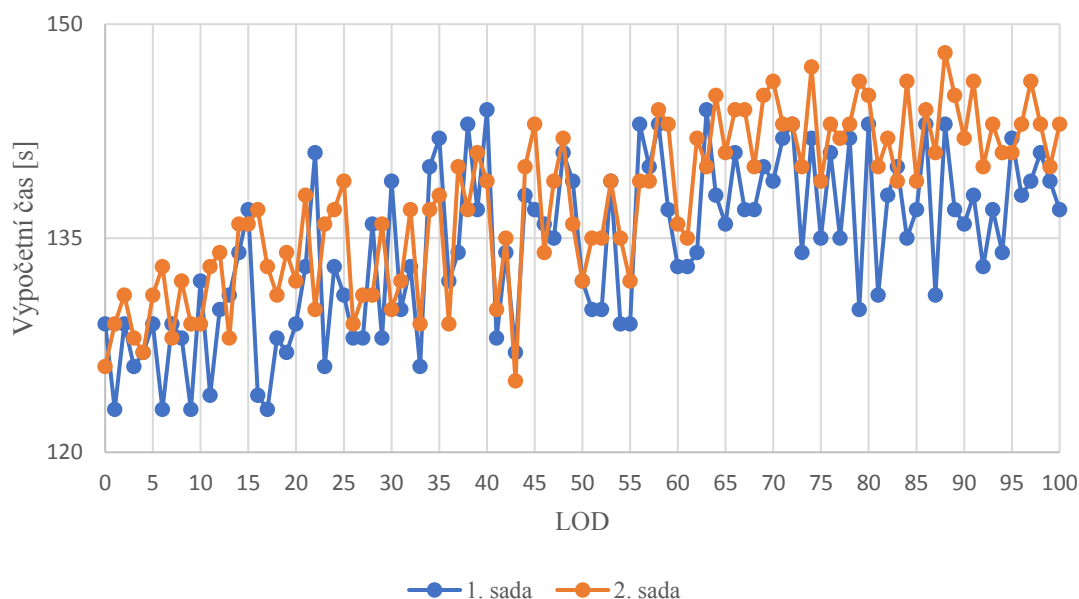
První sada projektů byla nakonfigurována s těmito parametry:

- Počet odrazů: 4
- Počet přenosů: 1
- Počet ohybů: 1

U druhé sady projektů byly tyto parametry změněny na tyto:

- Počet odrazů: 6
- Počet přenosů: 3
- Počet ohybů: 1

Výsledky prokázaly (viz obrázek 5.2), že výpočetní čas u druhé sady oproti první sadě vzrostl, ale ne o hodně. To odpovídá změně nastavení. Dále lze pozorovat, tak jako u první sady, že čím bylo LOD materiálů složitější, tím bylo zapotřebí více času pro výpočet simulace.



Obrázek 5.2: První sada simulací ve srovnání s druhou sadou simulací

Shrnutí

Na základě výše uvedených poznatků a grafů lze vyvodit, že různé LOD materiálů má na výkonnost softwarového nástroje Wireless InSite vliv, ale velmi malý. Je však zapotřebí si uvědomit, že byl simulovaný pouze jeden vysílač proti desíti přijímačům. V momentě, kdy by byl počet přijímačů navýšen, domnívám se, že potom by již časové rozdíly ve výpočtech s různými LOD byly o mnoho více viditelné.

5.2 Otestování citlivosti

Wireless InSite nabízí po výpočtu simulací mnoho výstupů. Pouze jeden z nich se však jevil jako velmi vhodný pro otestování vlivu různých LOD materiálů na citlivost softwarového nástroje Wireless InSite. Tímto výstupem byla impulzní odezva (anglicky *time of arrival*) rádiového kanálu navrženého systému ve Wireless InSite.

Vlivem vícecestného šíření signálu v daném rádiovém kanálu, které je způsobeno odrazy, ohyby či rozptyly, dochází k tzv. únikům. Únik má za následek, že během předem definovaného časového intervalu, kdy přijímač očekává konkrétní příchozí blok dat, dochází ke kolísání úrovně přijímaného rádiového signálu a jednotlivé složky signálu (jednoho elektromagnetického impulsu) dojdou na přijímač v různých časech (s různým zpožděním) a s různou fází. Kvůli tomu, že se zpoždění a útlum mění v čase náhodně, lze chování rádiového kanálu popsat pouze statisticky a výsledkem je potom časově proměnná impulzní charakteristika tohoto rádiového kanálů. [39]

Mým cílem tedy bylo zjistit, jak hodně různé LOD materiálů ovlivní přijímaný signál na přijímači. Tedy jak moc to ovlivní příchod jednotlivých složek signálu na přijímač a jak moc se bude měnit výkonová úroveň těchto jednotlivých složek.

Projekty byly navrženy s jedním vysílačem a 10 přijímači. Jelikož porovnat a popsat výsledky v závislosti na LOD materiálů u všech přijímačů vůči vysílači by bylo časově velmi náročné, vybral jsem si k porovnání jeden přijímací bod (Rx1) vůči vysílači (Tx) v závislosti na 5 různých LOD materiálů. Tabulka 5.1 ukazuje, co jsem porovnával.

Tabulka 5.1: Porovnání výsledků 5 projektů s různým LOD

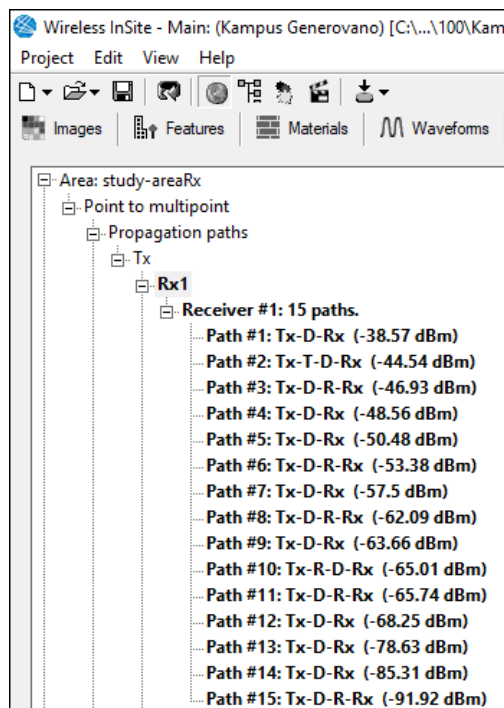
LOD	Materiál		Vysílač	Přijímač
	Beton [%]	Ostatní [%]		
0	100	0	Tx	Rx1
25	75	25	Tx	Rx1
50	50	50	Tx	Rx1
75	25	75	Tx	Rx1
100	100	0	Tx	Rx1

Porovnání výsledků jednotlivých simulací je k nalezení v podkapitole 5.2.2, nejdříve se podívejme na to, co vše lze zjistit z výsledků jedné konkrétní simulace

5.2.1 Ukázkový projekt

Pro představu u jednoho projektu ukážu výsledky přímo ze softwarového nástroje Wireless InSite v podobě obrázků, aby čtenář získal představu o tom, co vše lze simulacemi získat. Jedná se o projekt s LOD 0 – 100 % betonu a 0 % ostatních materiálů. K výsledkům, o kterých teď budu psát, došlo mezi vysílačem Tx a přijímačem Rx1, jak již jsem psal o něco výše.

Nejdříve si ukážeme, jak Wireless InSite zpracovává výsledky vícecestného šíření. Viz obrázek 5.3



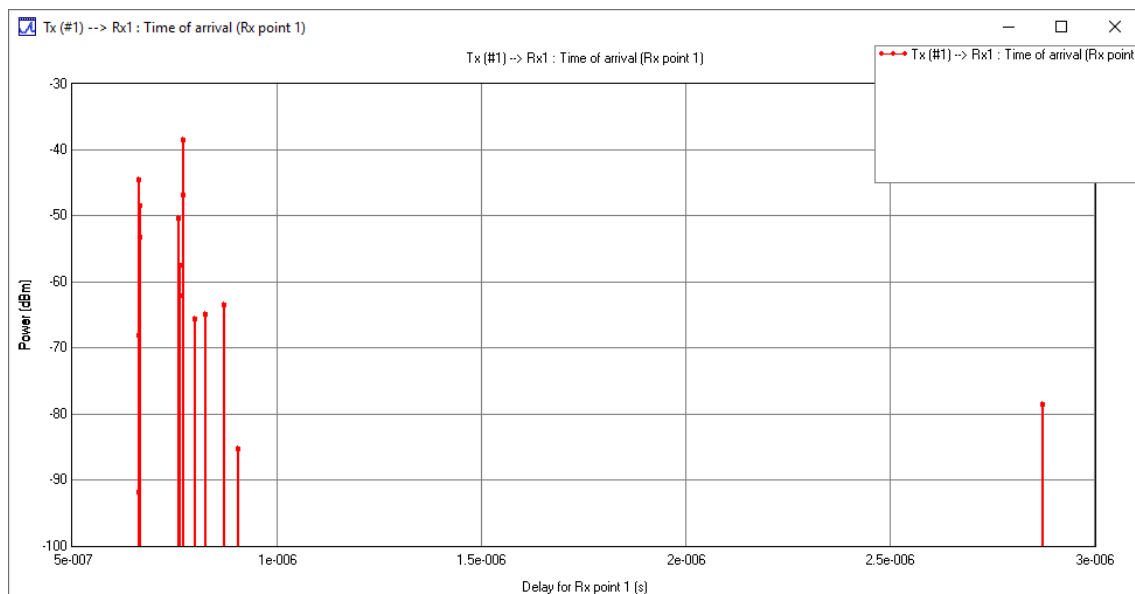
Obrázek 5.3: Zpracování vícecestného šíření softwarovým nástrojem Wireless InSite

Na obrázku lze vidět, že při šíření signálu ze zdroje (vysílače Tx) do cíle (přijímače Rx1) došlo k únikům. Tedy, jak již bylo vysvětleno v úvodu této kapitoly, signál dorazil s různým zpožděním, neboli jednotlivé složky signálu dorazily vlivem různých cest do cíle v jiných časech a s jinou (relativní) výkonovou přijímanou úrovní. Také u jednotlivých složek signálu můžeme pozorovat různé zkratky. Ty popisují cestu ze zdroje k cíli. Tabulka 5.2 rozebírá, co tyto zkratky znamenají.

Tabulka 5.2 Význam zkratk při vícecestném šíření signálu

Zkratka	Význam
Tx	Vysílací bod (Transmitter)
Rx	Přijímací bod (Receiver)
R	Odraz (Reflection)
G	Odraz od země (Ground bounce)
T	Přenos skrz překážku (Transmission)
D	Ohyb (Diffraction)

Výsledkem tohoto jevu, jak již také bylo vzpomenuto na začátku kapitoly, je časově proměnná impulzní charakteristika simulovaného rádiového kanálů. Ta lze vidět na obrázku 5.4



Obrázek 5.4: Časově proměnná impulzní charakteristika simulovaného rádiového signálu

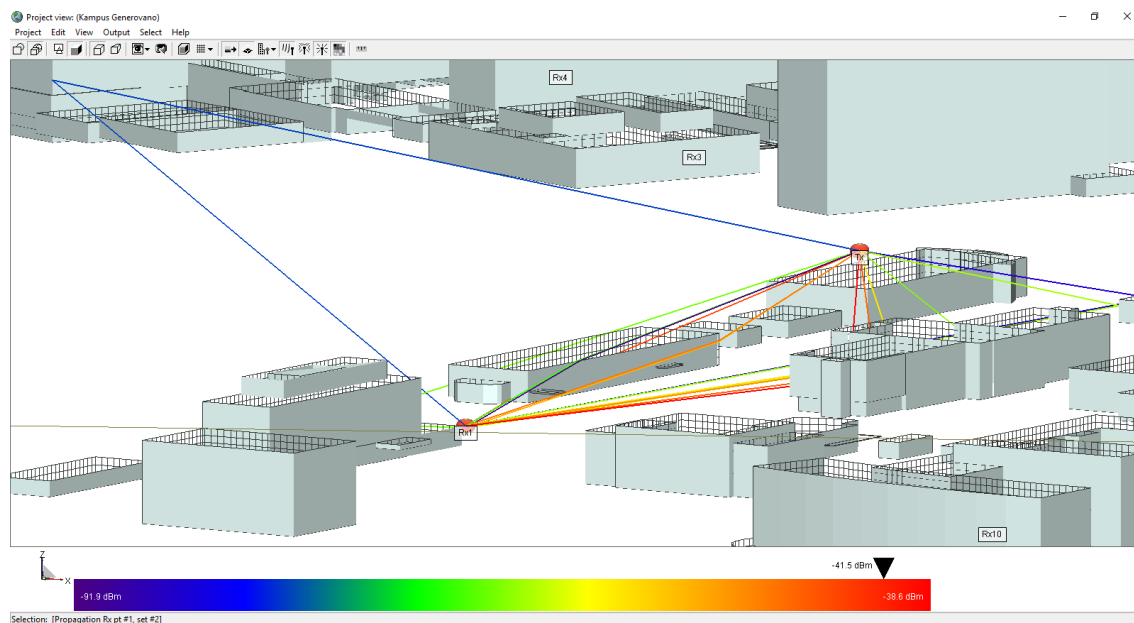
Aby šlo s daty dále pracovat, Wireless InSite je zpracovává do jednoduchého textového souboru. Viz obrázek 5.5

#	Receiver Set
1	15
1	7.72286e-07 -38.5689
2	6.63728e-07 -44.5431
3	7.73386e-07 -46.9292
4	6.68106e-07 -48.5622
5	7.60231e-07 -50.4846
6	6.68177e-07 -53.3783
7	7.63163e-07 -57.4983
8	7.63231e-07 -62.0886
9	8.72241e-07 -63.6642
10	8.27224e-07 -65.0105
11	8.01207e-07 -65.7409
12	6.64913e-07 -68.2519
13	2.87393e-06 -78.6344
14	9.05485e-07 -85.3074
15	6.65126e-07 -91.9219

Obrázek 5.5: Textový soubor obsahující 15 různých složek signálu, které dorazily na přijímač v různých časových okamžicích (v sekundách) s různou přijímací výkonovou úrovní v jednotkách dBm

Wireless InSite taky dovoluje zobrazit si některé zajímavé informace o vypočtených výsledcích. Jde například o nejvíce a nejméně utlumenou složku signálu, nejrychlejší a nejpomalejší složku signálu nebo střední hodnotu přijímané výkonové úrovně všech složek signálu.

Nyní se můžeme podívat na grafický výstup výsledků (viz obrázek 5.6).



Obrázek 5.6: Graficky znázorněné vícecestné šíření signálu

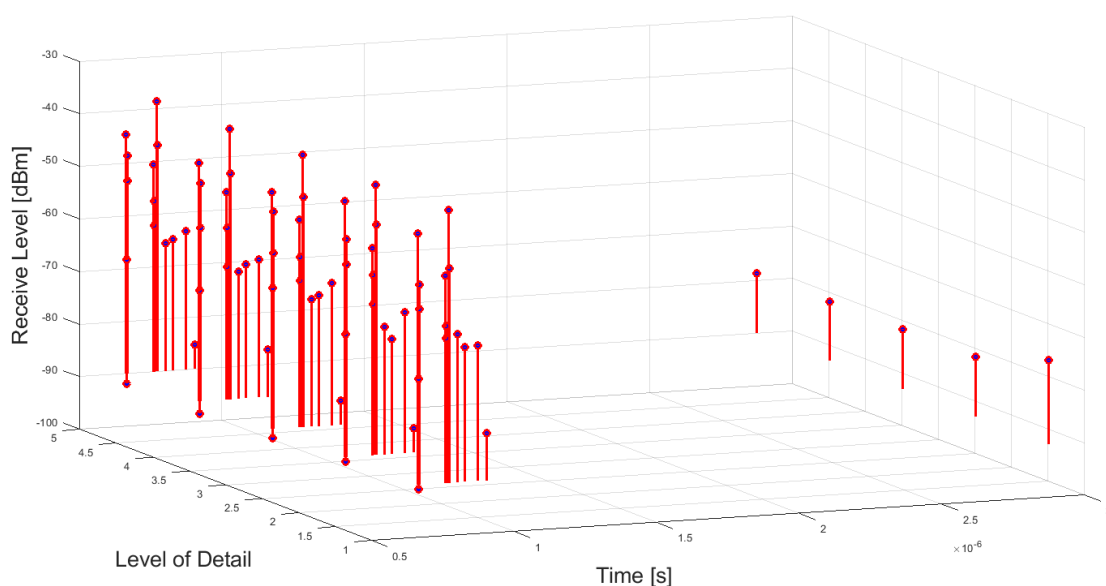
5.2.2 Porovnání výsledků

Tato podkapitola se zaměřuje na porovnání výsledků stejných projektů s různým LOD (0,25,50,75 a 100), viz tabulka 5.1. Porovnání jsem provedl prostřednictvím tabulky 5.3, kde porovnávám jednotlivé složky signálu mezi vysílačem a přijímačem (z hlediska času a výkonové přijímané úrovně) a poté data z tabulky porovnávám jako impulzní charakteristiky prostřednictvím grafu (obrázku 5.)

Tabulka 5.3 Porovnání jednotlivých složek signálu při různých LOD

LOD 0			LOD 25			LOD 50		
složka signálu	Čas [s]	Přijímaná výkonová úroveň [dBm]	složka signálu	Čas [s]	Přijímaná výkonová úroveň [dBm]	složka signálu	Čas [s]	Přijímaná výkonová úroveň [dBm]
1	7,72E-07	-38,5689	1	7,72E-07	-38,5689	1	7,72E-07	-38,1618
2	6,64E-07	-44,5431	2	6,64E-07	-44,5431	2	6,64E-07	-44,8079
3	7,73E-07	-46,9292	3	7,73E-07	-46,9292	3	7,73E-07	-46,202
4	6,68E-07	-48,5622	4	6,68E-07	-48,5622	4	6,68E-07	-48,5622
5	7,60E-07	-50,4846	5	7,60E-07	-50,4846	5	7,60E-07	-50,5089
6	6,68E-07	-53,3783	6	6,68E-07	-56,9661	6	6,68E-07	-56,5109
7	7,63E-07	-57,4983	7	7,63E-07	-57,2481	7	7,63E-07	-57,4983
8	7,63E-07	-62,0886	8	8,72E-07	-63,6642	8	7,63E-07	-62,0886
9	8,72E-07	-63,6642	9	8,27E-07	-64,5099	9	8,72E-07	-62,8795
10	8,27E-07	-65,0105	10	7,63E-07	-64,713	10	6,65E-07	-63,0585
11	8,01E-07	-65,7409	11	8,01E-07	-65,7409	11	8,27E-07	-65,0105
12	6,65E-07	-68,2519	12	6,65E-07	-68,8787	12	8,01E-07	-65,7409
13	2,87E-06	-78,6344	13	2,87E-06	-78,6344	13	2,87E-06	-78,6344
14	9,05E-07	-85,3074	14	9,05E-07	-81,016	14	9,05E-07	-85,3074
15	6,65E-07	-91,9219	15	6,65E-07	-92,3063	15	6,65E-07	-91,6077
Průměr	8,96162E-07	-61,37229333	Průměr	8,96162E-07	-61,51770667	Průměr	8,96162E-07	-61,1053
LOD 75			LOD 100					
složka signálu	Čas [s]	Přijímaná výkonová úroveň [dBm]	složka signálu	Čas [s]	Přijímaná výkonová úroveň [dBm]			
1	7,72E-07	-38,5546	1	7,72E-07	-38,0776			
2	6,64E-07	-41,2446	2	6,64E-07	-42,1862			
3	7,73E-07	-46,1808	3	7,73E-07	-49,0947			
4	6,68E-07	-48,5622	4	7,60E-07	-50,5073			
5	7,60E-07	-50,4846	5	6,68E-07	-51,9251			
6	6,68E-07	-53,3783	6	6,68E-07	-56,5109			
7	7,63E-07	-55,649	7	7,63E-07	-60,1443			
8	7,63E-07	-61,2401	8	8,01E-07	-61,7339			
9	8,72E-07	-63,2106	9	7,63E-07	-62,3886			
10	8,01E-07	-65,7409	10	8,72E-07	-64,1682			
11	6,65E-07	-66,5983	11	8,27E-07	-64,2814			
12	8,27E-07	-68,1318	12	6,65E-07	-69,7309			
13	2,87E-06	-78,6344	13	2,87E-06	-73,9			
14	9,05E-07	-85,3074	14	9,05E-07	-81,016			
15	6,65E-07	-90,7713	15	6,65E-07	-90,7713			
Průměr	8,96162E-07	-60,91259333	Průměr	8,96162E-07	-61,09576			

Jak lze vidět z přiložené tabulky, jednotlivé složky signálů jsou si velmi podobné. A to jak z časového hlediska, tak i z hlediska přijímané výkonové úrovně. Tento fakt také podporuje zprůměrování těchto hodnot, jak také ukazuje tabulka – i zde jsou výsledky téměř totožné. Z toho vyplývá, že LOD materiálů nemělo moc velký vliv na výsledky simulací. V některých případech jsou výsledky dokonce totožné, jindy se liší pouze v jednotkách dBm (co se týče přijímané výkonové úrovně) a v desítkách mikrosekund (co se týče času příchodu části signálu na přijímač z vysílače). Tyto výsledky graficky porovnává také následující graf. Jelikož jsem na osu LOD nemohl dát hodnoty 0,25,50,75 a 100, použil jsem hodnoty 1.6,2.6,3.6,4.6 a 5.6, kde 1.6 = LOD 0 a 5.6 = LOD 100.



Obrázek 5.7: Graficky znázorněné jednotlivé složky signálu

Shrnutí

Přiložený graf (obrázek 5.7) ukazuje, tak jako tabulka 5.3, že vliv LOD materiálů na přenášený signál byl minimální. Podobné výsledky vykazovaly i jiné interakce mezi vysílačem a přijímači a potvrdily jenom předchozí závěry.

Závěr

Cílem této práce, co se týče praktického hlediska, bylo v první řadě vytvoření algoritmu pro automatické generování struktur s různým LOD materiálů do formátu, který by byl schopen zpracovat softwarový nástroj Wireless InSite od firmy Remcom. K vytvoření tohoto algoritmu jsem využil jazyka Python a již vytvořený 3D model kampusu VŠB-TUO ve formátu .kmz a .object, poskytnutý katedrou geoinformatiky. Algoritmus byl navržen tak, aby byl schopen s určitou mírou pravděpodobnosti měnit materiály jednotlivých polygonů, což jsou základní geometrické útvary, z kterých se skládají objekty – v tomto případě 3D objekty budov. Výstupem tohoto algoritmu bylo 101 datových souborů, s různým LOD materiálů. Jen pro představu, 1. datový soubor obsahoval 0 % betonu a 100 % zbylých 8 materiálů, 51. soubor obsahoval 50 % betonu a 50 % zbylých 8 materiálů a 101. soubor obsahoval 100 % betonu a 0 % zbylých 8 materiálů

Materiálů, kterých jsem využil, bylo celkem 9: Beton (concrete), kov (metal), asfalt (asphalt), cihla (brick), sladká voda (fresh water), sklo (glass), dřevo (wood), podlahová deska (floor board), mokrá zemina (wet earth). Hlavními parametry těchto materiálů byla tloušťka, permitivita, konduktivita a drsnost povrchu. Hodnoty těchto parametrů byly vzaty z databáze materiálů zabudované přímo ve Wireless InSite. Stojí za zmínku, že tyto hodnoty byly převzaty z doporučení *ITU-R P.2040 [38]*, které se zabývá vlastnostmi stavebních materiálů pod vlivem frekvencí do 100 GHz.

Poté, co jsem měl k dispozici všech 101 souborů s různými LOD materiálů, vytvořil jsem ve Wireless InSite jeden projekt. To, jak jsem tento projekt nastavil je popsáno v kapitole 4. Po vytvoření projektu jsem otestoval jeho funkčnost s prvním LOD souborem a po zhlednutí výsledků jsem usoudil, že nastavení projektu půjde použít i při dalších simulacích. Rozkopíroval jsem tedy tento projekt na 101 projektů a do každého z těchto projektů jsem přidal unikátní LOD soubor.

Proto, abych nemusel každý projekt spouštět zvlášť, vytvořil jsem dávkový soubor pro Windows, který spouštění projektů jednoho po druhém z automatizoval a výsledky mi ukládal do mnou určených složek. Úkolem tohoto dávkového souboru bylo také zaznamenávání času potřebného pro provedení výpočtů simulací do externího souboru.

Když byly výsledky všech simulací hotové, začal jsem zkoumat vliv různých LOD materiálů na výkonnost a citlivost softvérového nástroje Wireless InSite. Nejdříve jsem se věnoval výkonnosti. Velkým plusem Wireless InSite je to, že dokáže pro zpracování výsledků využít technologie CUDA u grafických karet společnosti NVIDIA. Průměrná doba výpočtu jednoho projektu tak byla vždy kolem 3 minut, namísto nějakých 5 a půl hodin za použití procesoru. Ale zpět k vlivu různých LOD materiálů na výkonnost Wireless InSite. Výsledky prokázaly fakt, že různé LOD materiálů má na výkonnost softwarového nástroje Wireless InSite vliv, ale velmi malý. Ukázalo se, že projekty s jednodušším LOD materiálů byly spočítány dříve, i když ne o moc dříve. Naproti projekty se složitějšími LOD materiálů se počítaly déle.

Druhým zkoumaným parametrem byl vliv různých LOD materiálů na citlivost nástroje Wireless InSite. K otestování citlivosti byl vybrán výstupní parametr *time of arrival*, což lze do češtiny přeložit jako *doba příchodu signálu* (ze zdroje – vysílače do cíle – přijímače). Tento

parametr bývá častokrát nazýván také jako impulzní odezva rádiového kanálu. Po porovnání výsledků (impulzních charakteristik), jak formou tabulky, tak formou grafu, jsem byl nucen usoudit, že vliv různých LOD materiálů na citlivost použitého softwaru nebyl nijak zásadní. Výsledky, respektive impulzní charakteristiky simulovaných projektů s různým LOD byly skoro stejné až na pár jednotlivých složek signálu, které se od sebe lišily minimálně.

Závěrem lze tedy usoudit, že u softwarového nástroje Wireless InSite nemá moc velký smysl řešit, jestli bude 3D model simulovaného prostředí odpovídat skutečnosti po materiálové stránce. Výsledky totiž budou velmi podobné, ne-li stejné. Na druhé straně, pokud 3D model prostředí bude dostatečně propracovaný po materiálové stránce, časově to bude stále skoro stejně únosné, jako kdyby byl 3D model tvořen pouze jedním materiálem. Tento závěr však počítá s faktem, že se simuloval pouze jeden zdroj signálu (vysílač) vůči deseti přijímacím bodům. Bylo by tedy zajímavé, a tím by šla tato práce rozšířit, navrhnout projekt ve Wireless InSite tak, aby obsahoval několik tisíc přijímacích bodů rovnoměrně od sebe vzdálených a pak prozkoumat opět výkonnost a citlivost softwarového nástroje Wireless InSite. Časově by to zřejmě bylo dost náročné, ale možná by se ukázalo, že při větším počtu přijímacích bodů by snad různé LOD materiálu mohly mít větší vliv na již zmiňovanou výkonnost a citlivost nástroje Wireless InSite.

Použitá literatura

- [1] *Vektorový grafický formát DXF - Root.cz* [online]. [cit. 2018-11-3]. Dostupné z: <https://www.root.cz/clanky/vektorovy-graficky-format-dxf/>
- [2] *Specification | DXF File Format | Scan2CAD* [online]. [cit. 2018-11-3]. Dostupné z: <https://www.scan2cad.com/dxf/file-specification/>
- [3] *COLLADA - Wikipedia* [online]. [cit. 2018-11-4]. Dostupné z: <https://en.wikipedia.org/wiki/COLLADA>
- [4] *COLLADA Overview - The Khronos Group Inc* [online]. [cit. 2018-11-4]. Dostupné z: <https://www.khronos.org/collada/>
- [5] *COLLADA - COLLADA Public Wiki* [online]. [cit. 2018-11-4]. Dostupné z: <https://www.khronos.org/collada/wiki/COLLADA>
- [6] *Keyhole Markup Language - Wikipedia* [online]. [cit. 2018-11-6]. Dostupné z: https://en.wikipedia.org/wiki/Keyhole_Markup_Language
- [7] *KML Tutorial* [online]. [cit. 2018-11-6]. Dostupné z: https://developers.google.com/kml/documentation/kml_tut
- [8] *What is KMZ? What Opens a KMZ? File Format List from WhatIs.com* [online]. [cit. 2018-11-7]. Dostupné z: <https://whatis.techtarget.com/fileformat/KMZ-Google-Earth-Saved-Working-Session-file>
- [9] *Shapefile - Wikipedia* [online]. [cit. 2018-11-7]. Dostupné z: <https://en.wikipedia.org/wiki/Shapefile>
- [10] *Shapefile - Wikipedie* [online]. [cit. 2018-11-7]. Dostupné z: <https://cs.wikipedia.org/wiki/Shapefile>
- [11] *8.3 filename - Wikipedia* [online]. [cit. 2018-11-8]. Dostupné z: https://en.wikipedia.org/wiki/8.3_filename
- [12] *STLD – ICANNWiki* [online]. [cit. 2018-11-12]. Dostupné z: <https://icannwiki.org/STLD>
- [13] *STL (file format) - Wikipedia* [online]. [cit. 2018-11-13]. Dostupné z: [https://en.wikipedia.org/wiki/STL_\(file_format\)](https://en.wikipedia.org/wiki/STL_(file_format))
- [14] *Chuck Hull - Wikipedia* [online]. [cit. 2018-11-13]. Dostupné z: https://en.wikipedia.org/wiki/Chuck_Hull
- [15] *Stereolitografie – Wikipedie* [online]. [cit. 2018-11-13]. Dostupné z: <https://cs.wikipedia.org/wiki/Stereolitografie>
- [16] *The StL Format | fabbers.com* [online]. [cit. 2018-11-13]. Dostupné z: http://www.fabbers.com/tech/STL_Format
- [17] LUEBKE, David P. *Level of detail for 3D graphics*. Boston, MA: Morgan Kaufmann Publishers, c2003. ISBN 978-155-8608-382.
- [18] *Team Fortress 2 Is Too Detailed* [online]. [cit. 2019-1-3]. Dostupné z: <http://pressxordie.com/2011/01/24/team-fortress-2-is-too-detailed/>

- [19] *Level of detail* – *Wikipedia* [online]. [cit. 2019-1-3]. Dostupné z: https://cs.wikipedia.org/wiki/Level_of_detail
- [20] HUPKA, Dušan. *TECHNIKY „LEVEL OF DETAIL“ V KNIŽNICI OPENSCENEGRAPH*. Brno, 2014. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ.
- [21] *Level of detail* – *Wikipedia* [online]. [cit. 2019-1-8]. Dostupné z: https://de.wikipedia.org/wiki/Level_of_Detail
- [22] *Level of detail* – *Wikipedia* [online]. [cit. 2019-01-8]. Dostupné z: https://en.wikipedia.org/wiki/Level_of_detail
- [23] *Progressive meshes* – *Wikipedia* [online]. [cit. 2019-01-13]. Dostupné z: https://en.wikipedia.org/wiki/Progressive_meshes
- [24] KOLBE, Thomas Heinrich. *CityGML-OGC standard for photogrammetry?* [online]. 2009 [cit. 2019-01-14]. Dostupné z: https://www.researchgate.net/publication/285810926_CityGML-OGC_standard_for_photogrammetry
- [25] *SPATIO-SEMANTIC COHERENCE IN THE INTEGRATION OF 3D CITY MODELS* [online]. [cit. 2019-01-15]. Dostupné z: https://pdfs.semanticscholar.org/ea40/ce69aa23b1ce28d6ad742cb24277feb68281.pdf?_ga=2.30145468.22914389.1549221714-1881737665.1545205182
- [26] *CITYGML-BASED 3D CITY MODEL FOR ENERGY DIAGNOSTICS AND URBAN ENERGY POLICY SUPPORT* [online]. 2013 [cit. 2019-01-15]. Dostupné z: https://pdfs.semanticscholar.org/4f10/65cd7d29e9d81882de4dcb26d6083b8e2557.pdf?_ga=2.88847896.22914389.1549221714-1881737665.1545205182
- [27] *Siradel Smart City Explorer* - *Siradel* [online]. [cit. 2019-01-21]. Dostupné z: <https://www.siradel.com/software/smart-city-explorer/>
- [28] *Siradel 3D cities* - *Siradel* [online]. [cit. 2019-01-21]. Dostupné z: <https://www.siradel.com/3d-data/cities/3d-cities/>
- [29] *Siradel 3D districts buildings* - *Siradel* [online]. [cit. 2019-01-21]. Dostupné z: <https://www.siradel.com/3d-data/cities/3d-districts-buildings/>
- [30] *Wireless EM Propagation Software* - *Wireless InSite | Remcom* [online]. [cit. 2019-01-22]. Dostupné z: <https://www.remcom.com/wireless-insite-em-propagation-software>
- [31] *Wireless InSite Reference Manual: Version 2.8.0*. State College, PA 16801, October 2015
- [32] STRYJA, Lukáš. *Práce se softvérovým nástrojem Wireless InSite Pro* [online]. Ostrava, 2017 [cit. 2019-01-26]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/118913/STR0358_FEI_B2647_2612R059_2017.pdf?sequence=1&isAllowed=y. Bakalářská práce. VŠB - TUO.
- [33] *Geometrie/Raytracing* – *Wikiknihy* [online]. [cit. 2019-01-27]. Dostupné z: <https://cs.wikibooks.org/wiki/Geometrie/Raytracing>

- [34] YUN, Zhengqing a Magdy F. ISKANDER. Ray Tracing for Radio Propagation Modeling: Principles and Applications. In: *IEEE Access* [online]. 2015, **3**, s. 1089-1100 [cit. 2019-01-27]. DOI: 10.1109/ACCESS.2015.2453991. ISSN 2169-3536. Dostupné z: <http://ieeexplore.ieee.org/document/7152831/>
- [35] *Relativní permitivita - fyzikální tabulky* [online]. [cit. 2019-02-2]. Dostupné z: <http://www.converter.cz/tabulky/relativni-permitivita.htm>
- [36] *Electrical resistivity and conductivity - Wikipedia* [online]. [cit. 2019-02-2]. Dostupné z: https://en.wikipedia.org/wiki/Electrical_resistivity_and_conductivity
- [37] *Surface finish - Wikipedia* [online]. [cit. 2019-02-2]. Dostupné z: https://en.wikipedia.org/wiki/Surface_finish
- [38] *Effects of building materials and structures on radiowave propagation above about 100 MHz: Recommendation ITU-R P.2040-1* [online]. In: . Geneva, 2015, 07/2015, s. 32 [cit. 2019-02-3]. Dostupné z: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.2040-1-201507-I!!PDF-E.pdf
- [39] ŠVIRÁK, Ing. Martin. Ústav radioelektroniky FEKT VUT v Brně, Purkyňova 118. *Simulace rádiových únikových kanálů* [online]. 2002 [cit. 2019-04-10]. Dostupné z: http://www.elektrorevue.cz/clanky/02043/index.html?fbclid=IwAR1nzlaqHxy1lsmNhOGdRmc_nykXLem62ggRdEPUHxEoOq493wtuDO94cw4str